Quality in open data based digital service ecosystem

Anne Immonen

VTT SCIENCE 159

Quality in open data based digital service ecosystem

Anne Immonen

Thesis for the degree of Doctor of Philosophy to be presented with due permission for public examination and criticism in auditorium TS101, at University of Oulu, on the 29th of Sebtember 2017 at 12 noon.

ISBN 978-951-38-8557-1 (Soft back ed.)
ISBN 978-951-38-8556-4 (URL: http://www.vttresearch.com/impact/publications)

VTT Science 159

ISSN-L 2242-119X
ISSN 2242-119X (Print)
ISSN 2242-1203 (Online)
http://urn.fi/URN:ISBN:978-951-38-8556-4

JULKAISIJA – UTGIVARE – PUBLISHER

Teknologian tutkimuskeskus VTT Oy
PL 1000 (Tekniikantie 4 A, Espoo)
02044 VTT
Puh. 020 722 111, faksi 020 722 7001

Teknologiska forskningscentralen VTT Ab
PB 1000 (Teknikvägen 4 A, Esbo)
FI-02044 VTT
Tfn +358 20 722 111, telefax +358 20 722 7001

VTT Technical Research Centre of Finland Ltd
P.O. Box 1000 (Tekniikantie 4 A, Espoo)
FI-02044 VTT, Finland
Tel. +358 20 722 111, fax +358 20 722 7001

Juvenes Print, Tampere 2017

Abstract

To a growing extent, the software systems of today are provided as digital services distributed across networks, dynamically fulfilling the complex demands of consumers. As people have access to the Internet almost everywhere with the help of the mobile devices, such digital services are expected to be available when requested, and to provide services reliably and without any interruptions. Recently, the use of freely available data on the Internet has increased continuously in the context of digital services. This kind of open data has been identified as providing several benefits to service providers, such as new ideas, services, data-based contents, and confirmation in business decision making. Digital service engineering itself is evolving, and is shifting from isolated development environments towards open innovation and co-development environments, called ecosystems. Digital service ecosystems enable service providers to strengthen their position by cooperating, while still being able to act independently. The ecosystem supports the business models of its actors, also enabling the utilisation of existing ecosystem assets, such as knowledge and services.

This dissertation concentrates on the quality of digital service, with an emphasis on open data in ecosystem-based service engineering. The contribution of this research is a concept of an open data based digital service ecosystem, which provides the assets for service providers to design the quality of services and to ensure the quality of open data. These assets include the service engineering model that enables quality-driven service co-innovation and co-development among ecosystem members, the knowledge that can be utilised in digital service engineering, and the enabling environment with knowledge management models and support services for acting in the ecosystem. Additionally, the ecosystem provides support for defining an open business model, for evaluating the quality of open data, and for communication between digital service providers and open data providers. The ecosystem concept is generic, and can be adapted to different application domains; the domain model used together with generic knowledge management models adapts the service engineering and digital services, for example, to the healthcare, energy or traffic domains. The developed concept has been validated incrementally in several application domains.

Tiivistelmä

Yhä suurempi osa nykyisistä ohjelmistoista tarjotaan käyttäjille digitaalisina palveluina. Digitaaliset palvelut ovat tyypillisesti tietoverkkoihin hajautettuja palveluja, jotka vastaavat dynaamisesti palvelunkäyttäjien monimutkaisiin ja jatkuvasti muuttuviin vaatimuksiin. Koska ihmisillä on nykyisin pääsy internetiin kaikkialta, erityisesti mobiililaitteiden avulla, he olettavat näiden palvelujen olevan aina saatavilla sekä toimivan luotettavasti, ilman keskeytyksiä. Palveluntarjoajien kiinnostus avoimeen tietoon on viime aikoina lisääntynyt huomattavasti, ja avoimen tietoon perustuvia digitaalisia palveluja on alkanut ilmestyä markkinoille. Avoimen tiedon on huomattu tarjoavan paljon hyötyjä palveluntarjoajille, kuten uusia ideoita, palveluja ja dataan pohjautuvaa sisältöä, sekä vahvistusta ja tukea yrityksen päätöksentekoon. Digitaalinen palvelunkehitys itsessään on siirtymässä kohti avoimia innovaatio- ja yhteiskehitysympäristöjä, joita kutsutaan ekosysteemeiksi. Ekosysteemi tukee toimijoidensa liiketoimintaa ja tarjoaa myös tukea, kuten olemassa olevaa tietämystä ja tukipalveluja, joita eri toimijat voivan hyödyntää omassa toiminnassaan.

Tämä väitöskirja keskittyy digitaalisten palvelujen laatuun avointa tietoa hyödyntävässä digitaalisessa palveluekosysteemissä. Tutkimuksen pääkontribuutio on avoimeen tietoon perustuvien digitaalisten palvelujen ekosysteemikonsepti, joka tarjoaa tarvittavan tietämyksen ja aputoiminnot, joiden avulla digitaalisten palvelujen tarjoaja voi saavuttaa laatuvaatimukset ja varmistua myös palvelussa käyttämänsä avoimen tiedon laadusta. Konsepti sisältää laatukeskeisen palvelunkehitysmallin, joka mahdollistaa palvelun innovoinnin ja kehityksen yhdessä muiden ekosysteemin toimijoiden kanssa. Konsepti tarjoaa myös tietämyksen, jota voidaan hyödyntää palvelunkehityksessä, ja ympäristön, joka tarjoaa tietämysmallit ja tukipalvelut ja mahdollistaa niiden hyödyntämisen. Lisäksi ekosysteemi tukee siirtymistä avoimeen liiketoimintamalliin, tarjoaa tukea avoimen tiedon laadunvarmistukseen sekä mahdollistaa kommunikoinnin eri ekosysteemin toimijoiden välillä. Kehitetty konsepti on yleinen ja mukautettavissa eri sovellusalueille. Digitaalisten palvelujen kehitys voidaan mukauttaa esimerkiksi terveydenhoidon, energian tai liikenteen sovellusalueelle käyttämällä sovellusaluekohtaista mallia yhdessä yleisen tietämysmallin kanssa. Kehitetty ekosysteemikonsepti on varmennettu asteittain toteuttamalla osittaisratkaisuja eri sovellusalueiden ongelmiin.

Preface

The work presented in this dissertation was carried out in VTT Technical Research Centre of Finland Ltd, in several research projects. The work on digital services was implemented in ITEA's FAMILIES (FAct-based Maturity through Institutionalisation Lessons-learned and Involved Exploration of System), COSI (Co-development with inner and Open source in Software Intensive products) and ICARE (Innovative Cloud Architecture for Real Entertainment) projects, funded by VTT and Tekes – the Finnish Funding Agency for Innovation. The work on open data was performed in ODEP (Open Data End-user Programming) research project funded by Tekes, and in DIMECC's Need4Speed program funded by Tekes and VTT. I thank the mentioned institutions and my collaborators in VTT and other organisations and institutions for making this work possible. I am grateful to VTT for giving me the opportunity to work on various kinds of international and national projects and to achieve knowledge in a variety of topics, and finally to write and publish this doctoral dissertation.

Most of all, I thank my principal supervisor, Research Professor Eila Ovaska from VTT, who has over the years given me guidance and insight during the preparation of the original publications, and provided valuable support and comments for this dissertation. I also thank my other supervisor, Professor Veikko Seppänen from the University of Oulu, for providing specific thoughts and advice, and for helping me find the focus in the content of this dissertation.

I also thank the reviewers of this dissertation; Professor Kari Smolander from Aalto University, and Professor Matti Rossi from Aalto University, whose professional comments and suggestions improved this work significantly.

I am grateful to my other colleagues in VTT: Marko Palviainen, Jarmo Kalaoja, Daniel Pakkala, Pekka Pääkkönen and Tuomas Paaso, who participated in the research work and contributed to my original publications.

Finally, I thank my family; Marko, Matias and Aleksi, for their support and understanding while writing this dissertation.

Oulu, May 2017

Anne Immonen

Academic dissertation

Supervisors

Research Professor Eila Ovaska
Service and Information Architectures
VTT Technical Research Centre of Finland Ltd
P.O. Box 1100, FI-90571 Oulu, Finland
Adjunct Professor
University of Oulu
Faculty of Information Technology and Electrical Engineering
P.O. Box 3000, 90014 University of Oulu, Finland

Professor Veikko Seppänen
University of Oulu
Martti Ahtisaari Institute, Oulu Business School, and
Faculty of Information Technology and Electrical Engineering
P.O. Box 3000, 90014 University of Oulu, Finland

Reviewers

Professor Kari Smolander
Aalto University
Software Engineering, Department of Computer Science
P.O. Box 15400, 00076 Aalto, Finland

Professor Matti Rossi
Aalto University
Information Systems Science, Department of Information and Service Economy
P.O. Box 15400, 00076 Aalto, Finland

Opponent

Professor Tomi Männistö
University of Helsinki
Department of Computer Science
P.O. Box 68, FI-00014 Helsinki, Finland

List of publications

This dissertation is based on the following original publications which are referred to in the text as I–V. The publications are reproduced with kind permission from the publishers.

I Niemelä E., Immonen A. Capturing quality requirements of product family architecture. Information and Software Technology, 49(11–12), 2007. Pp. 1107–1120.

II Immonen A., Ovaska E., Kalaoja J., Pakkala D. A service requirements engineering method for a digital service ecosystem. Service Oriented Computing and Applications, 10(2), 2015. Pp. 151–172.

III Immonen A., Palviainen M., Ovaska E. Requirements of open data based business ecosystem. IEEE Access, Vol. 2, 2014. Pp. 88–103.

IV Immonen A., Pääkkönen P., Ovaska E. Evaluating the Quality of Social Media Data in Big Data Architecture. IEEE Access, Vol. 3, 2015. Pp. 2028–2043.

V Immonen A., Ovaska E., Paaso T. Towards certified open data in digital service ecosystems. Software Quality Journal (Published online: 21 June 2017), in press.

Contents

List of abbreviations

AaaS	Analytics-as-a-Service
API	Application Programming Interface
ATAM	Architecture Trade-off Analysis Method
CBSP	Component-Bus-System-Property
CDC	Connecting Digital Cities, Connecting Digital Cities/EIT ICT Labs project, one of the projects where this research results were tested
CKAN	Comprehensive Knowledge Archive Network
COSI	Co-development with inner and Open source in Software Intensive products, one of the projects where this research was performed
DaaS	Data-as-a-Service
DHR	Digital Health Revolution, one of the projects where this research was validated
DIKW	Data-Information-Knowledge-Wisdom
DIMECC	Digital, Internet, Materials & Engineering Co-Creation
DiSeP	Distribution Service Platform
EIT	European Institute of Innovation and Technology
EODE	Evolvable Open Data based digital service Ecosystem
EU	European Union
FAMILIES	FAct-based Maturity through Institutionalisation Lessons-learned and Involved Exploration of System, one of the projects where this research was performed
GQM	Goal-Question-Metric
IaaS	Infrastructure-as-a-Service

ICARE	Innovative Cloud Architecture for Real Entertainment, one of the projects where this research was performed
ICT	Information and Communication Technologies
IEE	Integrability and Extensibility Evaluation
INSPIRE	Infrastructure for Spatial Information in Europe
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
IT	Information Technology
ITEA	Information Technology for European Advancement
KMM	Knowledge Management Model
N4S	Need for Speed program, one of the projects where this research was performed
NFR	Non-Functional Requirements
NIMSAD	Normative Information Model-based Systems Analysis and Design
ODEP	Open Data End-user Programming, one of the projects where this research was performed
ODI	Open Data Institute
OSS	Open Source Software
PaaS	Platform-as-a-Service
PIR	Personal Information Repository
QADA	Quality driven Architecture Design and quality Analysis
QRF	Quality Requirements of a software Family
R&A	Reliability and Availability
R&D	Research and Development
RAP	Reliability and Availability Prediction
RE	Requirements Engineering
REST	Representational State Transfer
SBAR	Scenario-Based Architecture Reengineering
SLA	Service Level Agreement
SMEPP	Secure Middleware for Embedded Peer-to-Peer systems
Tekes	The Finnish Funding Agency for Innovation

TV	Television
UI	User Interface
UK	United Kingdom
UML	Unified Modeling Language
URI	Uniform Resource Locator
USA	United States of America
W3C	World Wide Web Consortium
WISA	Wireless Internet Service Architecture

1. Introduction

1.1 Background and motivation

A growing number of software systems are provided as digital services that are distributed across networks and dynamically fulfil complex consumer demands. People today are able to access the Internet from everywhere, especially with the help of their mobile devices, and require services that are rapidly discovered, ready to use, and that correspond to the individual needs of the consumer. This kind of setup causes the need for digital services that are discovered online, accessed through a well-defined interface, and controlled by the customer of the service (Chang and West, 2006). Thus, digital services can be anything that is entirely automated, and delivered digitally through an information infrastructure, such as the web, mobile devices, or communication networks.

An increasing number of today's digital services utilise data, that can be, for example, private data, public data, specially analysed data for the consumer, or integrated data from several sources to fulfil the consumer's needs. Recently, the interest in freely available, open data, was found to be high (Immonen, Palviainen and Ovaska, 2013). The concept of open data is based on the idea that certain data should be freely available for everyone to use and republish as they wish (Auer *et al.*, 2007). Open data can provide several benefits for digital service providers, such as new data based content, ideas and basic functions, increased understanding of business opportunities, improved competitiveness, potential new customers (Immonen, Palviainen and Ovaska, 2013), and insight into consumer opinions, preferences and requirements with regard to a company or its products/services (Antunes and Costa, 2012; Bhatia *et al.*, 2013; Fabijan, Holmström Olsson and Bosch, 2015). Thus, for example, real-time traffic data, weather data and maps can be provided as open data, which are then utilised by a service provider that integrates them and then provides a digital service targeted to travellers.

Competition among digital service providers is strong due to rapidly evolving markets, trends and customer needs. These challenges are a part of the reason why service providers have recently been shifting from isolated service engineering environments to more open innovation and co-development environments, called ecosystems. Digital ecosystems can be characterised according to Chang

and West (2006), being "open, loosely coupled, domain clustered, demand-driven, self-organising agents' environment, where each specie is proactive and responsive for its own benefit or profit". Species can include humans, economic species and digital species. The ecosystem allows its members to create value in networks flexibly and dynamically, following common regulations. In digital service ecosystems, services are co-innovated and co-developed by utilising common knowledge and existing ecosystem assets, such as design patterns, ontologies or analysis services. Open innovation (Chesbrough and Appleyard, 2007; Chan, 2013) enables companies to create ideas by themselves, use external ideas or co-create ideas with other companies of the ecosystem. The value networks and ecosystem infrastructure enable members to concentrate on their own roles and know-how, and to reach common goals. Value networks are formed by several organisations aiming to fulfil a certain purpose together (Allee, 2008; Lehto *et al.*, 2013), whereas ecosystem infrastructure (Khriyenko, 2012) manages the ecosystem's operations, making the services interoperable, available, and easily consumed. Therefore, the digital service ecosystem can be characterised as being an open, domain-clustered, demand-driven, self-organising and regulated environment, in which the actors of digital services co-innovate and co-create services in value networks, each having their own interests in services.

Since many digital services providing similar content or ideas exist in the market, quality of services has become a competitive advantage for service providers; digital services must embody high quality in order to guarantee customer satisfaction, and avoid problems that may cause serious financial and human safety-related damage and danger. This dissertation concentrates on the quality of digital services in the ecosystem context. Service consumers experience that service is of high quality when the service fulfils the consumers' task goals correctly, without interruptions, and is available when required. Thus, the quality can be defined as the probability of the system completing the tasks successfully when invoked; "reliability on demand" (Lyu, 1996). Quality has a broader meaning for service providers; the providers must be able to 'trust' that the service fulfils the business goals and the requirements, and works as expected. When the data becomes a part of a digital service, the quality of data also becomes part of the quality of services. The utilisation of open data requires proper knowledge about data quality; how reliable, trustworthy and valuable the data is for its intended use.

This dissertation takes the viewpoint of the service provider, having two quality focuses:

1) **How to design digital services in such a way that the services meet the quality requirements?** The digital services must be of high quality in order to be accepted and used by the consumers. To avoid extra costs, the quality requirements must be taken into account in early phases of the service engineering.
2) **How to ensure the quality of open data utilised in digital services?** The data must fulfil the quality requirements for its intended usage. The quality of data must be known before the data can be utilised.

This research can be positioned in the frontier of software engineering and data-intensive services. The usage of open data in digital services forces service providers to consider the transformation from the proprietary side of the software industry to a more open business model. Software engineering and data-intensive services collide in the case of digital services; servitization (Vandermerwe and Rada, 1988; Wiesner *et al.*, 2012) makes the open data more available for citizens through digital services. The quality of open data is emphasised when using it as part of a service; the quality-driven service design must also pay attention to data quality evaluation in order to enable the achievement of digital services with a high quality.

Quality analysis and evaluation in the case of software and services has a long history. Quality attributes, such as reliability and performance, are standardised (ISO/IEC, 2001), and several quality analysis approaches exist in the literature, such as the works of Cortellessa, Singh and Cukic (2002), Leangsuksun *et al.* (2003) and Reussner, Schmidt and Poernomo (2003). However, according to Immonen and Niemelä (2008), the existing approaches have several shortcomings. Most importantly, the approaches do not define how quality requirements can be transformed into different architectural decisions. Thus, there exists a huge gap between quality requirements and analysis, which also means that the traceability from quality requirements to quality analysis is missing. Quality must be engineered into software from the onset of its development; quality requirements must be used as a driving force in service engineering, and they must be analysed in an early phase, even prior to the service implementation, when fault corrections and modifications are easier and cheaper to perform, and design decisions can still be affected. In the case of digital services, the changing environment, changing user requirements and new quality concerns of consumers ensure the emergence of new service engineering models and methods, as well as dynamic quality evaluation methods. The service engineering model should introduce means for reaching the quality goals.

Quality and quality evaluation in the case of open data has not often been brought into use, although some standardisation efforts for data quality attributes exit, such as Data quality model (ISO, 2008a). The purpose of data quality evaluation is to ensure that the data fits its intended usage. Challenges in quality evaluation of open data are caused by unknown data sources, and by the growing amount of semi- and unstructured data, i.e. big data (Hashem *et al.*, 2015), with its features of large volume, variety, velocity and veracity (Ferrando-LIopis, Lopez-Berzosa and Mulligan, 2013). Additionally, service consumers have changed their expectations and perceptions of data quality in pervasive computing environments (Madnick *et al.*, 2009). The dynamicity, i.e. the changing usage environment, changing user needs, and dynamically changing situations in the markets (e.g. the data becomes suddenly unavailable), causes the need to continuously evaluate the data quality. Several definitions for data quality attributes exist (Wang and Strong, 1996; Nurse *et al.*, 2011), and some approaches have been introduced to achieve high data quality (Naumann and Rolker, 2000; Dai *et al.*, 2008; Nurse *et al.*, 2013). However, there still exists a lack of common agreement on the attrib-

utes and proper validation of existing quality evaluation approaches in the industry. Thus, the use of open data requires new evaluation methods that take these identified challenges into account.

In digital ecosystems, the business and the development environments are highly dynamic, and the needs and demands of service consumers are unclear and continuously changing. Therefore, although service engineering in a digital service ecosystem provides several business benefits, it also sets new challenges. Although some research has been carried out on service ecosystems (Liu and Nie, 2009; Riedl *et al.*, 2009; Ruokolainen, 2013), there is currently a lack of methods and approaches on how to take the digital service ecosystem elements into account in service engineering. Current research on service engineering and service ecosystems does not consider the quality viewpoint (the quality of digital services and quality evaluation of the data used in digital services). Therefore, new models are required for ecosystem based service engineering that enable the services to achieve their quality requirements. Furthermore, the ecosystem infrastructure that enables service co-development, cooperation and verification of quality of the data must be specified.

The characteristics of digital services and the utilisation of open data in services cause new challenges for delivering services and data reliably. This dissertation combines quality-driven digital service design and the evaluation of open data quality in a common digital service ecosystem context. The ecosystem's capability to perform actions should support quality-driven service engineering practices and data quality evaluation, with specific activities, knowledge models and support services. The main contribution of this dissertation is the concept of an evolvable open data based digital service ecosystem (EODE), which provides the assets to design the quality of services and to ensure the quality of open data.

1.2 Research objectives and questions

Based on the background and motivation described in the previous chapter, the main research question of this dissertation is **"How to design the quality of digital services in open data and ecosystem based service engineering?"** By designing the quality of digital services is meant all the models, methods, techniques and other means used in the service requirements specification and design phases that make it possible to engineer quality requirements, and transform them into architectural decisions and models of digital services.

This research has three objectives. The first is to research how service development has changed when shifting from the traditional, closed environment to more open ecosystems, and to investigate the main elements and phases for digital service engineering in the ecosystem. These elements and phases should enable the design of services of high quality.

The second objective is to investigate how quality evaluation has evolved when moving from the evaluation of software and services to data quality evaluation, and to understand the key phases for quality evaluation of open data. The purpose

is to describe how to ensure the quality of data, considering the identified challenges of open data and big data quality evaluation.

The third objective is to create a new model of a digital service ecosystem, in which the varying elements such as open (big) data, quality of data and dynamic digital service engineering, are considered in the assets provided by the ecosystem. Thus, the model combines the quality-driven, ecosystem-based service engineering and open data quality evaluation under a common ecosystem concept. The model specifies what kind of knowledge and services are required for engineering digital services that achieve their quality goals in the design phase and for evaluating the quality of open data in a way that is suitable for the purpose and the situation at hand. This certified data can then be utilised in digital services.

1.3 Scope of the study

In the following sub-chapters, the research themes of this dissertation are presented and the intersection of the main entities of the research is described.

1.3.1 Research themes

In this research, the focus is on digital service quality, and the quality of open data in the digital service ecosystem context, i.e. how to engineer reliable open data based digital services. Figure 1 describes the research topics of this dissertation. The quality of digital services in the ecosystem context comprises research themes for ensuring quality in service engineering and for certifying the quality of open data.

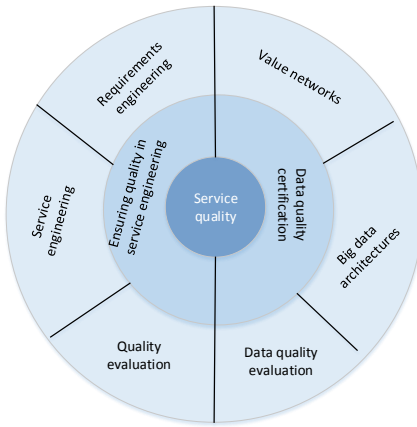

Figure 1. Research themes of this dissertation.

The theme of ensuring quality in service engineering is investigated through the themes of requirements engineering, service engineering and quality evaluation of services. Thus, the approach is methodology based, describing how to achieve quality in digital service engineering during the service engineering process. The existing quality evaluation approaches of services, especially their identified deficiencies (Immonen and Niemelä, 2008), provide the motivation and the starting point for the research. Current knowledge on service engineering (Erl, 2007; Sommerville, 2009) and requirements engineering (Kotonya and Sommerville, 1998; Nuseibeh and Easterbrook, 2000; Husnain, Waseem and Ghayyur, 2009; Loniewski, Insfran and Abrahão, 2010; Al-Fataftah and Issa, 2012) helps to identify and specify new service engineering model that takes take these requirements into account, enabling the achievement of quality by engineering the quality requirements, and capturing them into service architecture. The ecosystem context specifies common means and practises for the model to be applicable in ecosystem-based digital service engineering. In this research, the digital service quality is not limited to any specific attributes, and the application domain of the digital services is not limited, but the created concept is general and applicable to any service domain.

The theme of data quality certification is examined through the themes of value networks, big data architecture, and quality evaluation. In the case of open data, the quality cannot be inspected purely from the process viewpoint, since the quality of open data is highly affected by the usage context, i.e. how well the data fits its intended use (Wang and Strong, 1996; Nurse *et al.*, 2011). Thus, the approach to achieve data quality is specified more from the business viewpoint, starting from the business purposes of data utilisation. Data quality certification here means the confirmation that the open data is trustworthy, and its quality is good enough to be accepted for the use of the ecosystem's services. Trustworthiness (Nurse *et al.*, 2011) is achieved by data quality evaluation, proceeding from the evaluation of the data source and the data itself, to the evaluation of data for the usage context. Big data architectures offer the solution in the form of frameworks for handling huge amounts of data, providing a logical structure of core elements to store, access and manage big data (Ramesh, 2015). Big data architecture is also examined as a solution for handling the quality of data; the quality of data must be managed and controlled through business processes to be available in different contexts and situations. Value is formed in networks in the case of data (Kuk and Davies, 2011; Poikola, Kola and Hintikka, 2011), especially in the case of business ecosystems, when there is trust between ecosystem members. The data value networks, the data based business models (Perr, Appleyard and Sullivan, 2010; Teece, 2010; Tammisto and Lindman, 2011) and business ecosystems (Iansiti and Levien, 2004; Zhang and Fan, 2010; Li and Fan, 2011) provide the starting point for understanding open data based business. Although this research concentrates on open data, other kinds of data are also referred to. As data quality evaluation depends on several factors, such as the context, data type (open data, internal data), and the purpose of the data collection, quality attributes and quality evaluation must be adjusted according to the situation at hand. This research is not lim-

ited to any certain type of data, and the data quality is not limited to any specific attribute, although specific attributes that are applicable to open data are referred to.

Digital service ecosystem includes many activities, knowledge models, and support services, for example, for member cooperation, ecosystem governance, policy definition, and service modelling, implementation and testing. In this research, only the activities and elements that are related either to quality-driven service engineering, or data quality evaluation, are discussed in detail.

1.3.2 Intersection of main entities

The research described in this dissertation brings together three specific entities; digital services, open data and ecosystems. The intersection of these entities is described in Figure 2. The intersection of digital services and open data enables the creation of digital services that utilise open data. The intersection of digital services and the ecosystem forms a digital service ecosystem with the capability of engineering digital services utilising the ecosystem assets while obeying common regulations. The ecosystem ensures the service interoperability and provides an enabling environment for members for cooperation, and for supporting the business models of the members. The quality-driven digital service engineering must be enabled by the ecosystem knowledge management models and supporting services. Furthermore, the intersection of open data with an ecosystem forms an environment that enables the collaboration and cooperation of business actors of open data, also supporting business models of the members. The quality certification of open data becomes the responsibility of the ecosystem, which must provide means for data quality evaluation for its members. Finally, the intersection of all three intersections produces a cooperation environment that enables the quality-driven engineering of digital services utilising the quality certified open data.

In this dissertation, the main focus is in the middle of the intersections; the specified EODE concept describes how to perform quality-driven service engineering in a digital service ecosystem, and how to certify the quality of the open data utilised in digital services.

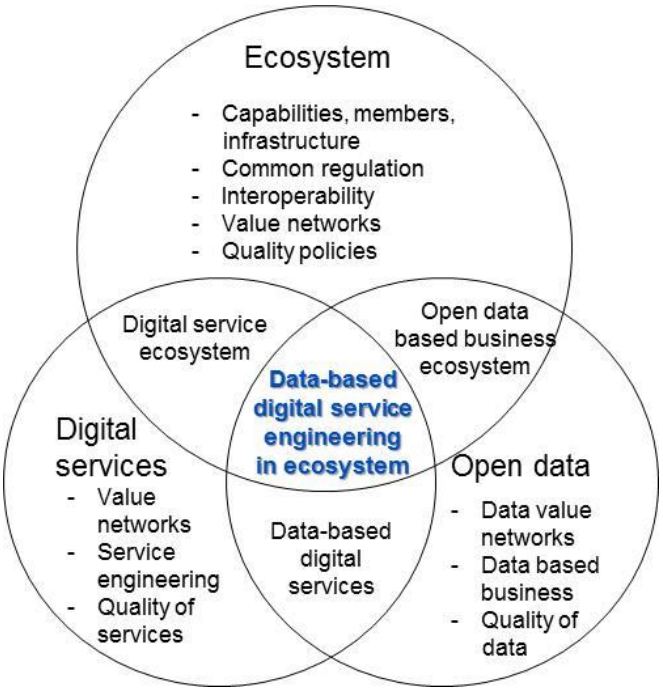

Figure 2. The intersection of the main entities of this research.

1.4 Research approach

A constructive research method (Järvinen, 2004) was selected for this dissertation, because this makes it possible to create solutions, i.e. constructs, to the identified problems, evaluate the applicability of the solutions, and reflect the findings back to the theory and practice of the problem areas. The constructive research method is applied in this research in two phases; first, it is applied in four separate research cases to provide solutions to the identified research problems, and then it is applied to combine and adapt the four specified solutions together, and to further extend the research in a larger problem area to provide a new concept of open data based digital service ecosystem. Figure 3 depicts how the research themes shown in Figure 1 were addressed during the research process. Publications of this dissertation are numbered in a logical order based on the research topics, and are presented as PI to PV in Figure 3. The research conducted was on two different topics: (1) The quality of services, and (2) the quality of data. The research included in this dissertation is summarised in Table 1.

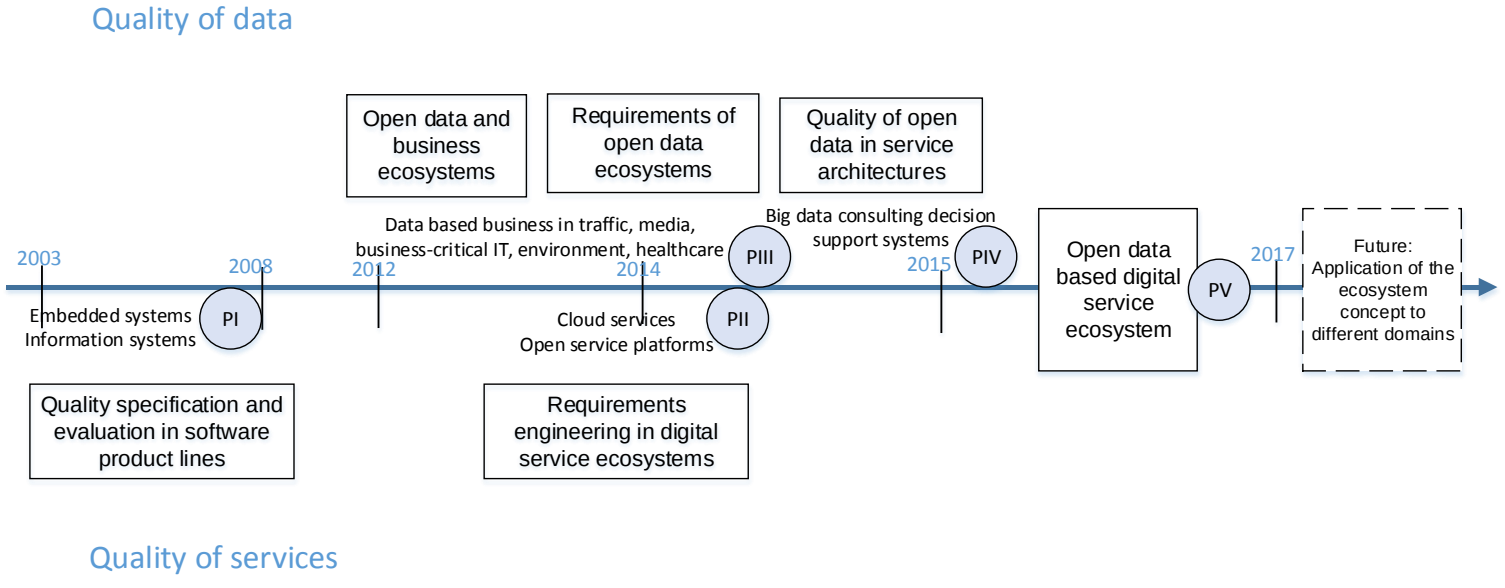

Figure 3. Progress of this research.

Table 1. Summary of the research conducted.

Publi-cation	Theme	Domain	Research description
I	Quality specification and evaluation in software product lines.	Distributed, embedded systems, and information systems in the context of product lines.	Development and validation of the method for capturing quality requirements in different kinds of application domains and product families.
II	Requirements engineering in digital service ecosystems	Service ecosystem; cloud services (producing content for multimedia services) and an open service platform (of multi-modal mobility services).	Development and validation of the RE method in real usage among two service ecosystems. Collecting users' feedback about the RE method.
III	Requirements of open data ecosystems.	Open data based business (environment monitoring, weather observation, media, healthcare, transport, UI design, mobile services, business-critical IT, data based services).	Requirements collection for the open data ecosystem from business actors in industry. Creation of the ecosystem concept, and its validation with involved industrial representatives.
IV	Quality of social media data in service architecture.	Decision support systems in business operation (in big data consulting), big data architectures.	Definition of the roles and responsibilities of business decision makers**,** and co-development of the solution for data quality evaluation with an industrial partner.
V	Quality management in open data service ecosystems.	General approach applicable to any service ecosystem domain.	Development and validation of parts of the concept in the context of different national research projects.

This research was conducted in several international and national research projects. The research on quality specification and evaluation in software product lines was started in the FAMILIES project in 2003.The author examined software reliability evaluation methods and approaches, and developed a framework for method evaluation in 2003–2005 based on a literature survey. In the COSI project, the author continued the research in software quality, and concentrated on quality-driven software development; more specifically on how to specify quality requirements of product families, and transform them into architectural models. The COSI project aimed to reduce software development costs, while achieving high quality using strategies to improve innovation for software-intensive product development. Research in the COSI project was conducted in 2005-2008, and resulted in a solution for quality requirement specification and transforming them to architecture; the Quality Requirements of a software Family (QRF) method. The QRF method was validated in 2007–2010, and was demonstrated to work in cases with

tens of quality requirements, across three different case studies. The case studies were: (1) the Personal Information Repository (PIR) case (Zhou *et al.*, 2011), (2) the Secure Middleware for Embedded Peer-to-Peer systems (SMEPP) case (Ovaska *et al.*, 2010), and (3) the DiSep platform case (Immonen, 2006; Immonen and Evesti, 2008). The QRF method has also been integrated and validated in connection with the Integrability and Extensibility Evaluation (IEE) method (Henttonen *et al.*, 2007).

Research on quality in software engineering proceeded from the context of product families to the context of service ecosystems, concentrating on requirement engineering in digital service ecosystems. Service engineering in digital service ecosystems was the main research focus of the author in the ICARE project. Research conducted on the FAMILIES and COSI projects could be utilised, since the product families could be seen as a kind of primitive software ecosystem, with features such as utilisation of existing assets, and a common knowledge base. The research concentrated on the features applicable to the ecosystem context, such as service co-innovation, service co-creation, enabling infrastructure and utilisation of the ecosystem's assets. The research was conducted in 2014–2015, and resulted in the concept of a digital service ecosystem, and a service engineering model. The results were validated in two different cases in the ICARE and CDC (Connecting Digital Cities) projects, where the project members acted as ecosystem members, co-developing digital services, and using the specified requirement engineering (RE) method. The ICARE project included 25 service ecosystem members from five European countries. Altogether, nearly 275 requirements were identified, including functional, non-functional and business requirements, and constraints. In the CDC project, seven European ecosystem members collected 23 requirements. The members also participated in the questionnaire about the RE method, which provided valuable feedback and information for the method refinement and development targets.

In parallel, the author began the research on open data and business ecosystems in the ODEP project, concentrating on identifying requirements of open data ecosystems. The author's main goal was to examine the business viewpoint of open data, and to outline the open data based business ecosystem from 2012–2014. The initial outline of the ecosystem was specified based on a comprehensive state-of-the-art literature survey, considering business ecosystems, data value chains, and business models of data. More specific requirements were collected, and the concept was validated in industry, when representatives from 11 different companies with different company sizes, application domains and service types provided valuable insight and refinements for the concept to respond to the actual needs of the data based industry, thus confirming the results of the literature research. The industrial representative interviews also helped to identify the motives and challenges faced when acting in open data ecosystems.

The unknown quality of data was identified as one of the major obstacles in open data utilisation. Therefore, the author continued research on open data in the N4S project, focusing on the quality of social media data in service architecture. In the N4S project, the author specified the solution to quality evaluation of open data

in big data architecture from 2013–2016. The solution used the identified challenges in data quality evaluation and the characteristics of big data as a starting point, and was based on current research on data quality policies, data quality attributes, and data evaluation techniques. The solution was partly developed with an industrial company; a big data consulting company, which also validated the solution with the help of trial usage. The solution provided the company with insight regarding customer needs, which could facilitate the company's own research and development (R&D).

The author continued research on open data, especially open data in service ecosystems, in the ODEP project in 2015–2016. Thus, two research branches, one concentrating on achieving quality requirements of services, and the other concentrating on evaluating the quality of data, were brought together. This merge formed an open data based digital service ecosystem, in which services are engineered in a way that enables them to capture quality in the design phase and utilise certified open data. The design focused on refining the service ecosystem elements such that they support data quality evaluation and management. The main ecosystem elements; the capability model, the knowledge models, and support services, were extended to include activities, models and services required for open data certification. New elements such as the ecosystem's core, and a data model for open data were also specified. The concepts of the ecosystem were validated incrementally in several research projects. The Digital Services Hub[1] was used as a core of ecosystem in the ICARE project in 2015. The project partners registered their services, and used the framework for authorising and visualising service connections. The semantic data model was developed in the Digital Health Revolution (DHR) project in 2015. The service data description enabled the link to commonly available schematics (e.g. schema.org or domain-specific ones), or optionally to a service-specific dictionary, and also enabled the data structure documentation to remain unchangeable.

1.5 Author's contributions

This dissertation consists of five original publications, which were published in 2007–2017. All the papers were published in peer reviewed scientific journals. The author of this dissertation is the first author in four publications, and the second author in one publication. The author's contribution to each publication is summarised below.

Publication I "Capturing quality requirements of product family architecture" describes the QRF (Quality Requirements of a software Family) method that specifies how quality requirements must be defined, represented and transformed to architectural models. This research is based strongly on the deficiencies in software quality evaluation methods identified based on an earlier literature survey of the author (Immonen and Niemelä, 2008). The specified method provides a solu-

[1] https://www.digitalserviceshub.com/registry/ (Accessed: 4 November 2016)

tion on how to engineer quality requirements, how to fill the gap between requirements engineering and architectural modelling, and how to trace requirements to architectural decisions and vice versa. The empirical evidence of the research was conducted across several case studies, where the method was applied and used in requirements engineering and architectural modelling of different kinds of cases. The author is the second writer of the publication, contributing to the state of the art literature survey, co-innovation of the method, and co-writing of the publication.

Publication II "A service requirements engineering method for a digital service ecosystem" describes the main requirements and elements for ecosystem based digital service engineering, and defines a service engineering model for digital service ecosystems, including the requirements engineering (RE) method. The publication also provides comparative definitions of the properties of the business ecosystem, service ecosystem and software ecosystem, and specifies the concept of a digital service ecosystem. The empirical evidence of the research was conducted by validating the RE method in two industrial cases, where the ecosystem members used the RE method for specifying digital services and related support services. Validation provided conformance of the method; it actually worked in a real case and also served as a valuable tool for communication among several people. The feedback collected from the method's users enabled the collection of user experiences, shortcomings, and development targets of the method. The author is the first writer of this publication, contributing significantly to the state of the art survey on business, service and software ecosystems, co-developing the service engineering model, specifying the ecosystem elements, and planning and executing the feedback collection from the users, and analysing the results.

Publication III "Requirements of open data based business ecosystem" specifies the first concept of an open data ecosystem from the business viewpoint. This ecosystem concept specifies the actor roles in the ecosystem, applicable business models, and the infrastructure, including the required support services and knowledge models for acting in open data based business. The outline and requirements of the ecosystem were collected based on the state of the art knowledge explored from the literature, and the state of the practice of data based business in the industry. Industry representatives provided valuable insights into the requirements of open data based business ecosystem from different stakeholder viewpoints. The author is the first writer of this publication, conducting the state of the art review on open data, open business models and open data ecosystems, planning and executing the interviews with co-authors and analysing the results, and co-specifying the new concept of the ecosystem.

Publication IV "Evaluating the Quality of Social Media Data in Big Data Architecture" specifies the elements and phases of quality evaluation of open data in big data architecture. The work is based on the identified challenges described in Publication III, where the unknown quality of the open data was detected as one of the main challenges for open data utilisation. The provided solution for quality evaluation enables the evaluation of the quality of open data for different contexts and situations with the help of data quality policies. The empirical part of this research was conducted by applying the method to an industrial case example to-

gether with industry representatives. The case provided the first proof of the usability of the suggested solution. The author is the first writer of this publication, contributing to the planning and execution of the interviews with colleagues and analysing the results, performing the state of the art review of relevant literature, specifying the data quality policies, and co-designing the solution for data evaluation.

Publication V "Towards certified open data in digital service ecosystems" combines, adapts and extends all the earlier work (Publications I–IV), and introduces the concept of evolvable open data based digital service ecosystem (EODE). The concept describes the main elements of the ecosystem, identifying the required actions, support services and knowledge models for digital service engineering and open data quality certification. The concept combines the viewpoints of open data providers and digital service providers; the ecosystem assists in verifying the quality of the data from different open data providers, and provides this data for service providers that utilise it in their services. The empirical part of this research was conducted in several cases, each case validating different parts of the concept. The author is the first writer of the publication, contributing to the scientific research by actively cooperating in the research planning and concept creation, and to the empirical part by analysing the results.

1.6 Structure of the dissertation

This dissertation consists of six chapters and the original publications. The first chapter provides a comprehensive introduction to the research work. The second chapter presents the theoretical foundation for the work; the main concepts and related work considering ensuring quality in service engineering, evaluating of the quality of open data, and ecosystem based service engineering. Chapter 3 summarises the research, describing how it was conducted, and introduces the results. Chapter 4 introduces the main research contributions based on the original publications. Chapter 5 provides a discussion about the accomplishment of the research objectives, evaluates the results from the theoretical and empirical viewpoints, evaluates the scientific validity of the research, compares the results to the related work and presents the limitations and future work. Finally, Chapter 6 concludes the dissertation. The original publications are provided as appendices at the end of the dissertation.

2. Theoretical foundation

This chapter discusses the theoretical foundation of the research. The first sub-chapter introduces the main concepts of this research, whereas the second describes work related to ensuring quality in software and service engineering, data quality evaluation, and ecosystem-based service engineering. Finally, the findings of the chapter are summarised.

2.1 Main concepts

In the following sub-chapters, the terms data, software and service are defined, and the concepts of open data, quality of open data, quality of software and services, and digital service ecosystems are discussed.

2.1.1 Data, software and service

According to Data-Information-Knowledge-Wisdom (DIKW) hierarchy (Ackoff, 1989), data is understood as symbols, and can be raw or processed. Raw data is produced by observing, monitoring, using questionnaires, etc., but is not yet processed for any specific purpose. Processed data is edited, cleaned or modified from raw data. Refinement and processing of data analyses, aligns and aggregates data from different physical and digital sources, increasing the understanding of the data, and thereby producing information from the data. Thus, information is data that is processed to be useful, providing answers to the questions who, what, where and when. Knowledge is created from data or information, referring to the theoretical or practical understanding of a subject. Theoretical knowledge represents explicit knowledge on the meaning of data, whereas practical knowledge is implicit and less systematically collected, represented and shared.

In general, software is a set of instructions or programs that instructs a computer to do specific tasks. Software is commonly divided into system software and application software. System software, for example device drivers, operating systems and compilers, controls the basic functions of a computer that are usually invisible to the user. System software is a base for application software, which can be a single program or a collection of small programs. Application software, such as office suites, gaming applications and database systems, handles the common

and specialised tasks a user wants to perform. Software can also contain data, for example libraries, different data structures, and some other forms of non-executable data.

In the context of software, a service is a piece of software developed to satisfy a need or to fulfil a demand. According to OASIS (2012), service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistently with constraints and policies as specified by the service description. Services are procured and paid for on demand, often requiring humans managing supplier-consumer relationships, allowing users to create, compose and assemble a service by bringing together a number of suppliers to meet needs at a specific point in time (Bennett *et al.*, 2000). A digital service is delivered digitally via the internet or other information network. The supply is essentially automated, or involves only minimal human intervention, and the service is always controlled by the customer. Digital service is implemented with software and can utilise different kinds of data, such as social media data, analysis data, marketing data, or any kind of data available through the information infrastructure.

2.1.2 Open data

Open data is data that it is freely available for everyone to use and republish as they wish, without restrictions of copyrights, patents or other mechanisms of control (Auer *et al.*, 2007). The concept of open data most notably has its roots in the UK, which has advanced the Open Government Data ecosystem during the past 15 years. The Open Definition of Open Knowledge in 2005 was recognised to be the first definition of open data. This definition has become and remains the key internationally-recognised standard: “Open data and content can be freely used, modified, and shared by anyone for any purpose”. A major breakthrough in the era of open data occurred in 2009, when both the UK and the USA launched their first data portals. Since then, the tendency in many countries has been that the data of the public sector collected along with tax revenues is obligated to be open. Many foundations and initiatives have been known to actively push organisations to open their data. Open Knowledge[2], founded in 2004, is a worldwide non-profit network of people that tends to unlock information, thus enabling people to work with it to create and share knowledge. Open Knowledge hosts a network of local groups in many countries and cities, in many different fields and topics. The organisation provides open source CKAN software, the world’s leading open-source data portal platform that publishes more than 1 million open datasets around the world. The Open Data Institute (ODI)[3] is a private limited company established as a non-profit organisation, limited by guarantee, and dedicated to promoting open data. It has member organisations all over the world. The Global Open Data Initia-

[2] https://okfn.org/ (Accessed: 29 September 2016)
[3] http://theodi.org/ (Accessed: 29 September 2016)

tive[4] aims to make Government data openly available to all, thus increasing awareness of open data, and supporting the development of the global open data community. The Initiative is led by civil society organisations to share principles and resources, and is meant to be used by governments and societies to learn how to best harness the opportunities created by opening government data. It is intended to provide a roadmap of policies and institutions that countries can use. The European Union (EU) has also begun to increase the utilisation of open data; the INSPIRE[5] (Infrastructure for Spatial Information in Europe) directive, enforced in 2007, is based on environmental spatial data infrastructure across Europe, established and operated by the 28 Member States of the EU. The EU also has an Open Data Portal[6] that provides a metadata catalogue, which enables the different parties of the EU to access to a great amount of data. The data in this portal adheres the Open Definition of Open Knowledge, being free to use, reuse, link and redistribute for commercial or non-commercial purposes.

Traditionally, open data has been provided by the public administration, as the opening of the administration data was considered to benefit business of the utilising companies, and thus the national economy, thus providing valuable information to the citizens. Recently, private companies have been interested in opening their own data, since open data has been identified as valuable, both in business and for a company's internal use (Immonen, Palviainen and Ovaska, 2013). The use of open data assists, for example, in information and knowledge-based management, and in decision making inside companies, in service development, and in data refinement. More recently, the amount of crowd-sourced information on the Internet has been continuously increasing, as people voluntarily produce new data that is available for everybody, or to certain social media groups. The different types of social media networks, such as Twitter, Facebook or Instagram, produce a significant portion of today's freely available data and information (Bodnar *et al.*, 2014). Companies are interested in this kind of data, since it has been known to provide several benefits for business, such as for predicting stock markets (Bollen, Mao and Zeng, 2011), or for analysing consumer reactions to specific brands (Jansen *et al.*, 2009).

At the same time, when the open data trend started to grow, data licences started to emerge. Several public licenses exist with which a licensor can provide access and copyright permissions to open the data, such as the Creative Commons License[7] and Conformant Licenses[8]. Licenses grant the baseline rights to distribute the copyrighted work, and most of the licenses still contain some elements restricting the utilisation of data, such as Attribution, Non-Commercial, No-Derivatives and Share Alike[9]. The different restricting elements can be mixed and matched, and therefore a huge number of customised licenses exist for open data.

4 http://globalopendatainitiative.org/ (Accessed: 3 October 2016)
5 http://inspire.jrc.ec.europa.eu/ (Accessed: 3 October 2016)
6 http://open-data.europa.eu/en (Accessed: 3 October 2016)
7 http://wiki.creativecommons.org/Baseline_Rights (Accessed: 20 September 2016)
8 http://opendefinition.org/licenses/ (Accessed: 20 September 2016)
9 https://creativecommons.org/licenses/ (Accessed: 20 September 2016)

Therefore, even if the data is freely available, its usage can be restricted with licences. Thus, the original idea of open data as such is no longer valid.

When utilising open data in business, the data becomes an artefact that is accessed by paying. In the data based business, open data does not adhere to the definition of open data, but can be thought of as open data services, that are provided to consumers. The consumers of the open data services are the digital service providers, providing digital services in which the open data service is used. Service related pricing models, such as pay-per-use (Weinhardt *et al.*, 2009) where the customer pays for the service usage, are therefore also applicable in the case of open data services. The final digital service consumer utilises open data through digital services.

2.1.3 Quality definitions of open data

Traditionally, companies have used their own data in business; i.e. the data collected from their own processes (e.g. production and warranty systems), from customer feedback, market analysis, or the data is bought from trustworthy third-parties. This data was assumed to be reliable. In the case of open data, the data comes from unknown sources, and the quality of the data was considered to be out of one's control. Therefore, more attention was paid to the quality of data.

Data quality has been traditionally characterised according to Wang and Strong (1996), as being data that is suitable for use by the data consumer. Data quality consists of quality attributes, which are a representation of a single aspect or a construct of a quality (Wang and Strong, 1996). Accordingly, the ISO/IEC data quality model (ISO, 2008a) specifies data quality as the "degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions". The ISO/IEC data quality model divides data quality into inherent quality, that refers to the intrinsic potential of the data to satisfy implied needs when used under specified conditions, and system dependent quality, that refers to reaching data quality within a computer system when data is used under specified conditions. The ISO/IEC data quality model specifies fifteen data quality attributes, such as accuracy, completeness, consistency, confidentiality and availability. In the literature, there exist several other classifications of data quality attributes, but there is no common agreement on their nature. Often, the attributes are too abstract, and there is a lack of agreed specifications and/or metrics for their evaluation. Traditionally, four quality dimensions are important for data consumers (Wang and Strong, 1996): Intrinsic dimension denotes the quality of data as independent of the user's context, whereas contextual dimension considers quality within the context of the task at hand, and the subjective preferences of the user. Representational dimension captures aspects relating to information representation, and accessibility dimension captures aspects involved in accessing information.

New data quality attributes are continuously required, as the origin and the nature of data change. Increasingly data comes from indeterminate sources, being commonly unstructured, or not more than semi-structured (Madnick *et al.*, 2009).

As the amount of this freely available data is enormous, the term “big data” is used to describe such massive volumes of data. Big data is data that is too difficult to process using traditional databases and software techniques, and its characteristics including volume, variety, velocity and veracity, cause new challenges for data quality and data quality evaluation (Ferrando-LIopis, Lopez-Berzosa and Mulligan, 2013; Cai and Zhu, 2015; Hashem *et al.*, 2015). Recent research on the quality of online data has been reviewed and summarised under three quality factors (Nurse *et al.*, 2011); provenance factors, referring to the source of information, quality factors, that concentrate on factors reflecting how an information object fits its use, and trustworthiness factors, that influence how end-users make decisions regarding the trust of information.

Several other works on on-line data quality attributes exist, such as Naumann and Rolker (2000), Gil and Artz (2007), Dai *et al.* (2008) and Nurse *et al.* (2013). Specific definitions of attributes have been conducted in the special case of on-line data, such as for social media (Agichtein *et al.*, 2008; Castillo, Mendoza and Poblete, 2011). In the work of Castillo, Mendoza and Poblete (2011), quality has been classified into credibility feature sets of social media data, including message-based features that consider characteristics of messages, user-based features that consider characteristics of the users who post messages, topic-based features that are aggregates computed from the previous two feature sets, and propagation-based features that consider characteristics related to the propagation tree that can be built from the retweets of a message. Furthermore, specific quality attributes exist, that are applicable to certain types of social media data. In particular, Twitter has recently interested researchers (Agichtein *et al.*, 2008; Castillo, Mendoza and Poblete, 2011; Ludwig, Reuter and Pipek, 2015). Chae (2015) classified the Twitter metrics into descriptive, content and network metrics.

Table 2 represents four different types of approaches to data quality and quality definition from the data consumers’ viewpoint. As can be seen from Table 2, there was quite a long time between the initial data quality definition (Wang and Strong, 1996) from the user’s viewpoint, and the data quality standardisation (ISO, 2008a). The data quality research concentrated first on internal data, such as production, sales, financial, and employee data. Since 2009, when the UK and the USA launched their first data portals, a lot of research has been conducted on the quality of open data, firstly on public administration data, and then on heterogeneous online data. Furthermore, the quality of online data was specified on a more detailed level, first for social media data, and then further for different types of social media data, such as Twitter (Castillo, Mendoza and Poblete, 2011). When concentrating on a certain type of data, the quality characteristics are also specialised at a detailed level, and may be applicable only to a certain data source type so that the data quality has a more specific meaning. Furthermore, in the case of online data, the number of quality properties is not limited but new specifications can be added as different kinds of data source types emerge.

It can be concluded that the quality evaluation of online open data is rapidly evolving, and new attributes and evaluation methods are constantly required. The

standardisation is slow, and therefore diverse approaches exist, and evaluation often occurs in an ad-hoc manner.

Table 2. Different definitions of data quality from the consumers' viewpoint.

	Traditional data quality (Wang and Strong, 1996)	**Data quality model** (ISO, 2008a)	**On-line data quality** (Nurse *et al.*, 2011)	**Twitter data quality** (Castillo, Mendoza and Poblete, 2011)
Year	1996	2008	2011	2011
Target of the approach	Data quality.	Data quality.	Data quality & trustworthiness.	Data credibility.
Data quality definition	Data that fits for use by data consumers.	The degree to which the characteristics of data satisfy stated and implied needs under specified conditions.	Fitness of information for use, and the perceived likelihood that a piece of information will preserve a user's trust in it.	Credibility in the sense of believability: "offering reasonable grounds for being believed".
Quality representation	Dimension.	Attribute.	Factor.	Feature.
Classification of quality properties	Intrinsic, contextual, representational and accessibility.	Inherent data and system dependent data.	Data provenance, quality, and trustworthiness.	Message-based, user-based, topic-based and propagation-based features.
The number of quality properties	20.	15.	Not set.	Not set.
Target data	Production and storage data.	Software product data.	Online data (social media data).	Twitter data.
Rationale	Important data dimensions for evaluation from data consumers' viewpoint.	Attributes for software product quality requirements and evaluation.	Factors of online data quality evaluation for data users.	Feature sets for Twitter data assessment for data users.

2.1.4 Quality of software and services

Quality in the case of software and services means the non-functional properties of the software, embodied as quality attributes. Digital services are implemented with software, usually consisting of software and data. Therefore, referring to soft-

ware functionality or a set of software functionalities, the standard quality attributes (ISO/IEC, 2003a, 2003b) can be used and applied in their evaluation. According to the ISO/IEC standard (ISO/IEC, 2005), a quality attribute is “an inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means“. A lot of work has been done in standardising quality attributes among software engineering (ISO/IEC, 2001, 2003a, 2003b). The ISO/IEC quality model (ISO/IEC, 2001) characterises quality attributes into functionality, reliability, usability, efficiency, maintainability and portability, each having several sub-characteristics. Quality is measured using quality metrics, which are measures of certain properties of the quality attribute, evaluating the degree of presence of the quality attribute (ISO/IEC, 2001). According to the ISO/IEC quality model (ISO/IEC, 2001), the software lifecycle is divided into three main phases, and quality attributes and metrics are divided into the same categories as the lifecycle. Internal quality is measured and evaluated from the design artefacts, such as architecture, design, and source code, against internal quality requirements. External quality is the quality when the software is executed, typically measured and evaluated while testing in a simulated environment, with simulated data. Quality-in-use is the user’s view of the quality when the software is used in a specific environment, and in a specific context of use, measuring the extent to which users can achieve their goals in a particular environment.

Quality analysis occurs both in the design phase and on run-time, and can be both static and dynamic (Immonen and Pakkala, 2014). In traditional software engineering, quality analysis has been an independent task performed after system implementation. By analysing quality before the implementation phase, time and resources can be saved. This predictive analysis enables the problems in quality to be solved more easily, at the architectural level, when modifications are easier and cheaper to implement. However, analysis of the architecture is only possible if it is represented in a way that enables the analysis (Jazayeri, Ran and van der Linden, 2000). Therefore, quality must be engineered into software from the onset of its development. The analysis approach must enable the derivation of quality requirements to architectural decisions in order to evaluate how the specified requirements are addressed in the architectural models. In addition, the analysis approach must enable the tracking of quality analysis results from the architectural models to the requirements in order to validate whether or not the requirements are met in the architecture (Immonen and Niemelä, 2008).

Consequently, quality is evaluated with the help of quality analysis methods that can be roughly classified into quantitative and qualitative. Quantitative methods, such as described by Smidts and Li (2000), Reussner, Schmidt and Poernomo (2003) and Pham and Defago (2013), are usually measurement based, quantifying systems already in use (black-box techniques) or making predictions considering a system’s internal structure (white-box techniques). Qualitative analysis methods, such as in the works of Chung *et al.* (2000) and Kazman, Klein and Clement (2000) manipulate knowledge rather than numbers. This knowledge is usually specific for the system under study, and can be explicit, i.e. documented or tacit, undocumented. Quality analysis methods of software have been developed over

40 years. The first quality analysis methods applicable at the architectural level emerged in the early 2000s when enhanced modelling methods were applied to architecture design. During the past decade, a transition from software and software architectures to service architectures has been observed. At the same time, quality analysis methods targeted specifically to services, such as (Grassi, 2004; Cortellessa and Grassi, 2007; Ma and Chen, 2008), started to emerge when the run-time quality management of services became important. The services were mostly functional and they were based on own (usually sensor) data, the quality of which was assumed to be good. However, the new kind of digital services discussed in this dissertation are based on open data from multitudinous sources. Therefore, the problem of service quality as discussed in this dissertation is new; the quality of data from uncertain sources becomes a part of the quality of the digital services.

2.1.5 Digital service ecosystems

There exist three different ecosystem definitions that are related to each other; business ecosystem, service ecosystem and software ecosystem.

- *Business ecosystem* is a dynamic structure of organisations that work together in a specific core business (Iansiti and Levien, 2004), sharing the common ecosystem regulation, being still able to act independently. The value is created in a network of actors, but apart from business dependencies, there are no dependencies between ecosystem members.
- *Service ecosystem* is a socio-technical system that enables service providers to reach shared goals, and gain added value by utilising the services of other members in the ecosystem (Liu and Nie, 2009; Riedl *et al.*, 2009; Ruokolainen, 2013). In service ecosystems, the focus is on dynamic, behaviour, and conceptual interoperability (Pantsar-Syväniemi *et al.*, 2012), and interactions between services, and between humans and services. Members share service taxonomy and service descriptions that can be categorised, for example, by domain, purpose or technology.
- In *software ecosystems* there is some common technology underpinning the ecosystem (Bosch, 2009; Jansen and Cusumano, 2012), when the focus is on technical, syntactic and semantic interoperability and interactions between systems and humans. Therefore, in software ecosystems there is an increased dependency between ecosystem members, although the features such as self-regulation, networked character and shared value are still valid.

Service ecosystems can be positioned between business and software ecosystems, filling the gap between the two. *Digital service ecosystems* are a part of service ecosystems, covering only the digital part. The product of a digital service ecosystem is a digital service that is entirely automated, available online, delivered through an information infrastructure, and controlled by the customer of service. Thus, for example, when comparing with traditional healthcare service ecosystems, the digital service ecosystem can provide devices and applications as ser-

vices used by a medical team, but the whole treatment process (including doctors, nurses, etc.) is not provided as a service.

Breaking the boundaries around a company is the first step when shifting towards ecosystem-based digital service development. Digital service ecosystems are an open cooperation environment, where companies can create ideas and develop services by themselves, use external ideas, or co-innovate and develop services with other actors of the ecosystem. Shifting to a more open business model may be required in order to understand the new business opportunities which the ecosystem provides. Open business models enable companies to break the boundaries around the company in the innovation or service engineering phase (Chan, 2013), or when delivering services. Seven open business models have been identified within the context of open-source software (OSS) (Perr, Appleyard and Sullivan, 2010), that can be classified into four categories based on how they capture value (Chesbrough and Appleyard, 2007): The deployment category includes support, subscription and professional services/consulting business models, which are similar to the proprietary side of the software industry. The hybridisation category includes proprietary extensions and dual-license business models, attempting to attract customers by licensing to familiarise the customers with the product/service. In the complements business category, open source software is provided by the vendor selling and supporting the hardware device or appliance, whereas in a self-service business model, users with similar needs pool their resources and create applications to satisfy the community's needs.

The ecosystem infrastructure provides the required support for service co-development, and actors' co-operation in the ecosystem, enabling utilisation of the existing ecosystem's assets. In service ecosystems, cooperation takes place in value networks. Service value networks provide business value through the agile and market-based composition of complex services from a pool of complementary service modules by the use of ubiquitously accessible information technology (Blau *et al.*, 2009). The ecosystem infrastructure provides the knowledge management model to guarantee the effectiveness of the service ecosystem by maximising semantic interoperability and alignment among ecosystem members, services and technologies. In addition, the knowledge base (Ovaska *et al.*, 2010) is required as a repository for storing the gained knowledge, such as collaboration models, service descriptions, ontologies, styles and patterns, which are then exploited in each service engineering phase.

2.2 Related work

In the following sub-chapters, the existing methods and approaches for ensuring quality in software and service engineering, data quality evaluation and ecosystem-based service engineering are discussed and summarised.

2.2.1 Ensuring quality in software and service engineering

The survey of reliability and availability (R&A) prediction methods at the architectural level (Immonen and Niemelä, 2008) revealed that the current analysis methods have several shortcomings. Most importantly, the methods do not commit themselves to quality requirements at any level, thus failing to define how the requirements could be transformed into different architectural decisions. In software engineering, the requirements engineering, architecture modelling and quality analysis phases are closely related. The architecture design phase is the first stage in which it is possible to evaluate how well the quality requirements are being met. To enable this, the architecture must be presented in a way that enables analysis (Jazayeri, Ran and van der Linden, 2000).

Several quality analysis methods have been available since the 1970s. The methods already applicable at the architectural level have been developed during recent decades. These methods are meant for different types of purposes, and have been developed by different communities. Accordingly, they have different definitions and measures for different quality attributes, as well as for architecture, inputs, outputs, notations, assumptions, users, etc. An analysis of eight quality evaluation methods was represented by Dobrica and Niemelä (2002), which appeared to verify that the quality requirements have been addressed in the architecture. According to the analysis results, the methods had different goals, such as guiding the architecture inspection, focusing on potential trouble spots, risk assessment, evaluation of the architecture to reach the software quality requirements, prediction of a certain quality attribute of a software system based on its architecture, or location and analysis of trade-offs in architecture. The research concludes that a multi-attribute analysis is required for understanding of the strengths and weaknesses of complex systems. Some of the surveyed methods were applicable for multiple quality attributes, such as the architecture trade-off analysis method ATAM (Kazman, Klein and Clement, 2000), and scenario-based architecture reengineering (SBAR) (Bengtsson and Bosch, 1998).

A comparison of design methods applicable for product line architecture was introduced by Matinlassi (2004). Four of the five methods surveyed ensured quality attributes with non-architectural evaluation methods, such as model checking, inspections and testing. Only one method, Quality-driven Architecture Design and quality Analysis (QADA) (Matinlassi, Niemelä and Dobrica, 2002), was able to evaluate software architecture designs before implementation. The QADA is a quality-driven architecture design method that uses quality requirements as the driving force when selecting software structures. The work of Dobrica and Niemelä (2000) defined different architecture views, and described the mapping of quality attributes to these views. QADA utilises these views, and represents the architecture design in conceptual and concrete abstraction levels, both of them consisting of structural, behavioural, deployment and development views. Quality attributes are mapped to the relevant views in QADA. The QADA method was first adopted to wireless service engineering (Niemelä, Matinlassi and Lago, 2003), where it

was detected to improve architectural descriptions and increase understanding of the meaning of service architecture.

Requirements engineering (Kotonya and Sommerville, 1998) has a long history, since incomplete, incorrect, and ambiguous requirements are generally considered to be the major cause of software failure (Dorfman and Thayer, 1997). Most of the RE methods concentrate on functional requirements, having several shortcomings. For example, RE methods lack tool support, do not cover all the phases of RE, or only present a technique applicable in a certain RE phase (Husnain, Waseem and Ghayyur, 2009). Several requirements engineering methods exist that consider quality requirements; the i* framework (Chung, Gross and Yu, 1999) helps to detect where the quality requirements originate, and what kind of negotiations should take place; the NFR (non-functional requirements) framework (Chung *et al.*, 2000) derives the quality requirements as goals from stakeholder needs, and uses them as guidance while considering different design alternatives; and the CBSP (Component-Bus-System-Property) method (Grünbacher, Egyed and Medvidovic, 2003) aims to reconcile requirements and architectures using intermediate models that are used as a bridge while refining and transforming the requirements to architectural elements. Architectural styles and patterns assist in mapping between quality requirements and architecture design, whereas for architectural modelling, an extension of a standard notation is required to avoid the development of an enormous amount of separate annotation and extension techniques. A standard and widely accepted modelling language, the Unified Modelling Language (UML) (OMG, 2002) has been extended by specific profiles to support quality attributes (Rodrigues *et al.*, 2003; Aagedal *et al.*, 2004; Cortellessa and Pompei, 2004). The standard UML can also be annotated with quality related information, such as using defined quality properties stored as stereotypes in the UML profiles (Ovaska *et al.*, 2010). Although some work had been done to engineer quality requirements and to represent quality in the architectural models, knowledge on how requirements affect architecture design has been missing.

Knowledge-based service engineering was first presented by Niemelä, Kalaoja and Lago (2005), when the QADA was adopted towards wireless service engineering with the help of a Wireless Internet Service Architecture (WISA) knowledge base. The WISA knowledge base provides the service taxonomy, the reference service architecture, and the WISA basic services. Further knowledge based service engineering is represented in the work of Ovaska *et al.* (2010), in which a quality aware modelling approach is defined as having three main phases; modelling quality requirements, representation of quality in architectural models, and model-based quality evaluation. Quality knowledge is utilised in all the phases in the form of quality attribute ontologies, model artefacts (i.e. styles and patterns), and reusable usage profiles. The knowledge base is further used in an IEE method (Henttonen *et al.*, 2007) that utilises the knowledge gained in architectural design.

Figure 4 describes the observed development direction of ensuring quality in the software and service engineering. The years in the figure describe the time when the topic was detected to be relevant, and when the first research on the

topic was conducted. Traditionally, testing has been a common procedure to ensure that software fulfils its requirements. Since testing finds quality problems only after the implementation phase, the prediction of quality in an early phase, at the architectural level, was found to save time and resources. However, quality prediction was not possible if the quality requirements were not represented in the architecture in an appropriate way. This was the driving force to quality-driven architecture design. Requirements engineering was finally considered to be important, since the requirements had to be engineered in a way that enabled evaluation of whether they were being met. Therefore the significance of knowledge in service engineering has increased when proceeding from testing to requirements engineering.

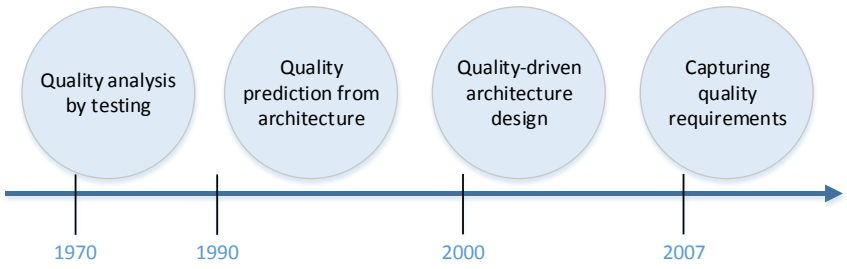

Figure 4. Evolution of quality assessment in software engineering.

2.2.2 Quality evaluation approaches of open data

From the first definitions of open data (Open Knowledge in 2005), it took about 5 years before the reusability of data from the user perspective was considered. The government of the UK proposed a five star scheme that helped assess the degree to which individual datasets were reusable (HM Government Cabinet Office, 2012). Recently, data certification approaches have emerged. For example, the Open Data Institute (ODI) provides Open Data Certificates[10] that enable data providers to assess the extent to which open data is published according to recognised best practises. The certificate tells data users what the data is about, and how to get hold of it, sharing legal (e.g. licensing, privacy), practical (e.g. discovery), technical (e.g. structure, quality) and social (e.g. documentation) information. Data certificates help to develop a shared understanding of open data; the certificate allows data providers to assess their own work (Heimstädt, Saunderson and Heath, 2014). Currently, both of these approaches rely on the information obtained from the data provider. Especially, information about the quality of the data may not be trustworthy as such, as in addition to being subjective, the data quality attributes, such as relevancy, cannot be judged to be valid in all situations.

[10] https://certificates.theodi.org (Accessed: 1 November 2016)

The quality evaluation of open data has been the subject of several studies (Naumann and Rolker, 2000; Agichtein *et al.*, 2008; Dai *et al.*, 2008; Castillo, Mendoza and Poblete, 2011; Nurse *et al.*, 2013). Some data quality evaluation approaches have emerged to utilise and extend the ISO/IEC data quality model. In the work of Behkamal *et al.* (2014) a metric-driven approach for quality assessment of linked open data was proposed, based on ISO 25012. The approach identifies five new inherent quality characteristics, and introduces a Goal-Question-Metric (GQM) approach applied to evaluation of all the characteristics. In the work of Rafique *et al.* (2012), two more characteristics are added to the ISO 25012 data quality model, and other characteristics from the ISO 25012 standard and from proposed previous research are grouped, and a framework is proposed to specify quality requirements for information of web applications.

The importance of quality policies has been recognised in several works. In the works of Bizer (2007) and Bizer and Cyganiak (2009), a quality-based information filtering policy is suggested, that consists of a set of metrics for assessing quality dimensions that are relevant for the task at hand, and a decision function that aggregates the resulting assessment scores into an overall decision as to whether information satisfies the consumer's quality requirements. In the work of Chenyun *et al.* (2009), a confidence policy denotes subjects to whom the policy applies, why certain data are accessed, and the minimum level of confidence that has to be assured by the data covered in the policy, when the subjects to whom the policy applies need to access the data for the purpose specified in the policy. In the framework of Bertino and Lim (2010), trust scores are associated with all data items to indicate their trustworthiness. Trust scores can be used for data comparison or ranking, or used together with other factors (e.g., information about contexts and situations) to decide about the use of the data items. The framework utilises the confidence policy of Chenyun *et al.* (2009) to specify the range of trust scores that a data item must have for use by the application or task. The approach described by Rahman, Creese and Goldsmith (2011) uses an information source filter to subscribe to a set of known information sources, and a scoring function to capture the provenance factors of interest, and to assign scores to messages for each factor. The decision making policy allows the decision maker to amplify or attenuate one or more provenance factors that may appear to be more or less important in a particular situation. The work of Mendes, Mühleisen and Bizer (2012) introduces a framework in which user-selected metadata is leveraged as quality indicators to produce quality assessment scores through user-configured scoring functions.

The four different types of approaches that exist for the evaluation of data from the user perspective are summarised in Table 3, introducing both the apparent benefits and the potential pitfalls of the approaches.

Table 3. The benefits and potential pitfalls of the different types of data quality evaluation approaches.

Approach type	Benefits	Potential pitfalls
Checklists, schemes, certificates (HM Government Cabinet Office, 2012); (https://certificates.theodi.org)	• Rapidly accessible information about the data. • Trustworthy data provider; the provider must be registered.	• Subjectivity of information. • Variation in data quality description; data quality can be described at a high abstraction level.
ISO/IEC data quality model (ISO, 2008a)	• Standard and well-known quality attributes.	• No guidance for the evaluation of quality attributes (only an example for measurement). • Slowly extendable for new types of data (e.g. social media data)
Stand-alone data quality evaluation approaches (Naumann and Rolker, 2000; Agichtein et al., 2008; Dai et al., 2008)	• Often solves the problem of quality evaluation for the case at hand.	• Not easily adaptable to different situations. • Isolated; not working together with other approaches. • Immature; not applied in the industry.
Policy-based approaches (Bertino and Lim, 2010; Rahman, Creese and Goldsmith, 2011)	• Can be configured to different situations. • Is rapidly adaptable to changes. • Easily extendable.	• Immature; not applied in industry.

As shown in Table 3, the different types of approaches provide some benefits, but have several shortcomings that complicate their usage. In order to be valid, data quality cannot be estimated by the data provider. Therefore, for example, the data certificate must be provided by a third party. The ISO/IEC data quality model (ISO, 2008a) can only be thought of as a guideline that describes which attributes the data quality consists of. The problem with standards is that they evolve slowly, whereas new quality attributes are constantly required for evaluating the rapidly emerging data and data sources. The different stand-alone evaluation approaches are often limited to the solution of a specific problem, and their application to different contexts is difficult. Data quality policies describe the principles and guidelines used to manage and exploit data, and information resources. Configuring them makes it possible to reach the needs of the specific data consumer. However, it can be concluded that data evaluation approaches are not mature; their practical application in the industry is missing. Most of all, none of them are applied in the context of ecosystems.

2.2.3 Ecosystem-based digital service engineering

No methods or approaches considering digital service engineering in service ecosystems were found in the literature. This may be due to the fact that the concept of digital service ecosystem is relatively new, and although the level of interest in service ecosystems is high, no common ecosystem regulations, capabilities or knowledge models have been specified. In general, the ecosystem requires an enabling cooperation environment for different actors to co-innovate and co-develop digital services together. Additionally, ecosystem infrastructure should enable the utilisation of common assets, such as knowledge and other services.

Several approaches exist that enable open service innovation. Several differences can be detected between them:

- Many of the approaches, such as Riedl *et al.* (2009) and Chan (2013), use the *central platform* to support cooperation, extract ideas for service innovation, and attract businesses and citizens to create e-services based on available data.
- The approach of Chesbrough and Appleyard (2007) uses the *underlying architecture* to connect different pieces of innovation components, and considers the value proposition for different partners.
- A *process model* of the innovation framework of Stathel *et al.* (2008) groups innovation activities into five main functions required to develop new services. The functions are idea mining, idea development, idea evaluation, service realisation and service evaluation.
- The effect of the *value chain* in innovation, requirements engineering performance and software success, is a major contribution of the work of Fricker (2010).
- The effect on *socio-technical aspects*, such as context, environment, and team management in service innovation is emphasised by Schindlholzer, Uebernickel and Brenner (2011).

The literature has only few suggestions on how to go further from service innovation to service co-creation. The Inter-enterprise Service Engineering Framework (Kimita *et al.*, 2009) supports three phases of e-service development in business ecosystems: requirements analysis, service design and service implementation, assigning them to strategic, conceptual, logical and technical abstraction layers. Service requirements are identified in the strategic perspective in the form of a business model. In the work of Wiesner *et al.* (2012), guided questionnaires were used to elicit the requirements coming from the current business situation, and a workshop was held to define the basic requirements for each Manufacturing Service Ecosystem scenario. The work of Stathel *et al.* (2008) included a mapping of information collected in the Innovation Repository accessible to service engineering, but the approach did not describe how this information affects service realisation.

Interoperability models provide the enabling infrastructure where the loosely coupled services can collaborate. A few classifications can be found from the literature:

- Six *interoperability levels* are defined for smart environments (Pantsar-Syväniemi *et al.*, 2012), including conceptual, behavioural, dynamic, semantic, communication and connection levels.
- Four *inter-related metamodels* are suggested for ecosystem interoperability (Ruokolainen and Kutvonen, 2009), including domain ontology, methodology, domain reference, and knowledge management metamodels.
- *Pragmatic interoperability* (Ruokolainen and Kutvonen, 2009) is achieved between ecosystem members when their intentions, business rules and organisational policies are compatible.

Knowledge- and ontology-based requirements engineering has a long history (Dobson and Sawyer, 2006), and even currently, an increasing amount of research has been conducted to utilise ontologies in RE (Castañeda *et al.*, 2010). Different kinds of approaches have been suggested, such as an approach for generating a requirements model based on the concepts in the service requirements modelling ontology (Xiang *et al.*, 2007), or establishing a mapping between a requirements specification and ontological elements (Kaiya and Saeki, 2005). In the work of Ovaska and Kuusijärvi (2014) the reusable artefacts, such as ontologies, models, patterns and rules, were provided in the knowledge base. An approach for developing intelligent applications/services for smart spaces was introduced by Ovaska, Salmon Cinotti and Toninelli (2012) and Pantsar-Syväniemi *et al.* (2012), that exploited the ontology models, interoperability models and context models for describing self-adaptable services. The approach was multi-technology and multi-domain oriented, but still lacked the business (ecosystem) viewpoint.

2.3 Summary of findings

According to the literature review, there already exist several methods and approaches that enable some parts of ecosystem-based digital service engineering. A new infrastructure is required to support the activities of service providers. Furthermore, the responsibility of the infrastructure is expanded when combining the activities of the open data actors, under the same ecosystem context. Several features could be identified for the digital service ecosystem that combines the open data provider's and digital service provider's viewpoints, and also considers quality in both of these viewpoints. These are described in Table 4.

Table 4. The features of an open data based digital service ecosystem.

Feature	Description
1. Service co-innovation model.	Open innovation breaks the boundaries around a company in the innovation phase; either the components of external knowledge and innovation are used in service development, or a company allows external parties to use its knowledge and innovation components in service development (Chan, 2013). The ecosystem must provide a model that enables the co-innovation of services utilising the assets of the ecosystem.
2. Service co-development model.	In an ecosystem, the digital service engineering must be transformed to a new model. Different actors, existing knowledge, existing digital services and open data all have an influence on digital service engineering. A new kind of service engineering model must be able to take these issues into account.
3. Knowledge based service engineering.	Existing knowledge is a valuable asset for all ecosystem actors. The ecosystem must provide a model of how knowledge can be utilised in businesses of the ecosystem actors, in engineering digital services and in providing open data.
4. Enabling infrastructure.	The enabling infrastructure makes services interoperable, available, and easily consumable. The ecosystem must provide an infrastructure to manage all service ecosystem operations including the ecosystem's regulation and management, and the support for the activities of digital service providers and open data providers.
5. Open business model.	Traditional business models concentrate on gaining profits by overtaking competitors and keeping strict boundaries around the company. The ecosystem forces companies to re-think their business strategies and models, as in an ecosystem, the business cannot be shut down within the boundaries that surround the company. Transformation to an open business model (Perr, Appleyard and Sullivan, 2010) requires a lot of investment, and newly assessed business model elements. The ecosystem must assist each actor in finding its business model and providing support for the business model elements.
6. Quality evaluation of open data.	Data quality evaluation is challenging due the fact that the data quality cannot be judged without considering the context at hand (Wang and Strong, 1996; Nurse *et al.*, 2011). The purpose, importance, and the type of data in the situation at hand determine how the data is to be evaluated. The ecosystem must provide means to evaluate and manage the quality of open data that can be then utilised by the ecosystem members.
7. Support for cooperation of the actors of open data and digital services	Governance and regulation actions are required for different actors to find their place and to cooperate in the ecosystem. The ecosystem must provide support for actions such as finding reliable partners, making contracts between ecosystem members, specifying SLAs and supporting bidirectional communication between digital service providers and open data service providers.

The current shortcomings of the state-of-the-art review considering the requirements of Table 4 are summarised as a result of this chapter, according to the following:

1. Service co-innovation: Several service innovation approaches exist, but they are separated from the further phases of service engineering. Furthermore, they do not take quality of services into account. Some requirements engineering methods consider quality requirements, but do not specify how to transfer them into architecture. Thus, currently no service requirements engineering methods exist that are applicable to co-innovation in the ecosystem context.

2. Service co-development: Quality has been taken into account systematically in many works dealing with software architecture design and analysis (Dobrica and Niemelä, 2002; Immonen and Niemelä, 2008; Gorton and Klein, 2015). The whole chain from requirements engineering to quality analysis must be specified in order to enable the capturing of quality requirements into the architecture, and to fill the gap between quality requirements and analysis results. Few approaches exist on how to transfer from co-innovation to co-development. Only a few methods enable quality-driven architecture design, but these do not consider the ecosystem viewpoint.

3. Knowledge-based service engineering: Some approaches have been presented that provide support for knowledge-based service engineering, that are prerequisites for service engineering in an ecosystem. These enable the utilisation of knowledge in the form of ontologies, including quality ontologies, in service engineering. These do not, however, consider business (ecosystem) viewpoints.

4. Enabling infrastructure: Currently, the open data is utilised in an ad-hoc manner. For open data, several communities, data portals and platforms exist that publish data and increase the awareness of data, but these do not assist in data utilisation in digital services. Some interoperability models exist that provide the enabling infrastructure in the form of interoperability models for service collaboration, but these are not applied in the ecosystem context. The required ecosystem elements have been defined from the ecosystem engineering viewpoint, but not yet from the digital service engineering viewpoint.

5. Open business models: Digital service ecosystems have recently emerged, and they have been detected to provide several advantages to service providers, such as collaborative innovation and value co-creation among ecosystem members, enabling the service providers to strengthen their forces. However, it is obvious that an open data business requires a transformation from the proprietary side to a more open business model. Currently, the open data business lacks business models and new operation models (Immonen, Palviainen and Ovaska, 2013).

6. Quality evaluation of open data: The quality of open data is often assessed by the data provider, when the information about the quality is subjective. Open data certificates of the ODI provide only general quality information about the open data. An evaluation of whether the data fits its intended use must be conducted by the data consumer. Several evaluation approaches exist for open data. These are not, however, easily adaptable to dynamic needs or applicable to the ecosystem context. Data quality certification must be provided by the ecosystem level that assists the data consumers to verify that the data source is trustworthy, the data adheres to current best-practices and its quality meets the ecosystem's requirements.

7. Support for cooperation of actors of open data and digital services: Open data communities and ecosystems operate separately from service ecosystems. No specified model or concept exists on how these actors must cooperate and share the same ecosystem assets.

In summary, the identified methods and approaches for ecosystem-based service engineering are loose. They only concentrate on their own viewpoint, and not on working together. Clearly, there is a lack of methods on how to take digital service ecosystem elements into account in service engineering. There are multiple actors, viewpoints and capabilities in the ecosystem that affect service engineering in an ecosystem, but the current approaches do not take these into account. Most notably, the quality of services is not taken into account in any of these approaches.

In the case of data, quality issues are not commonly brought into use, although a lot of work has been done to evaluate different quality attributes. The challenges in quality evaluation could be handled with data quality policies, that are used to generate quality objectives, also serving as a general framework for action (ISO, 2008b). Although some promising policy-based approaches already exist for quality evaluation, their practical application is has not been demonstrated. In addition, they are not present in the ecosystem context.

According to the state-of-the-art review, no ecosystem concept currently exists that combines the viewpoints of the open data provider and the digital service provider under the same ecosystem context. Furthermore, the quality of digital services or the open data has not been examined in the ecosystem context, which is the main focus of this dissertation. Currently, there exists no ecosystem concept that provides all the features of Table 4. This dissertation aims to create a new model of digital service ecosystem, which fulfils the identified features of Table 4, combining ecosystem-based service engineering and open data quality evaluation under the same ecosystem concept, and supporting the cooperation of different actors. In this dissertation, the ecosystem elements are specified from the viewpoint of digital service engineering, describing the knowledge and services required for engineering digital services that achieve their quality goals and for evaluating the quality of open data to be used in digital services.

3. Summary of the research

This chapter describes the research approach, method and process adhered to in this dissertation, and the achieved results and their evaluation. Finally, the research activities are summarised.

3.1 Research approach

The research approaches can be roughly divided into inductive and deductive (Trochim, 2006). Deductive research is a top-down approach, testing existing theories, and moving from the general to the more specific, whereas inductive research is a bottom-up approach, aiming to generate new theories, and thus moving from the specific to more general (Creswell and Plano Clark, 2007). This research follows the inductive approach. Inductive research generally consists of the following steps (Figure 5): data gathering, pattern development and theory development (Blackstone, 2012). In this research, the relevant data is first collected in different application domains. Patterns and regularities are detected from data that is collected and analysed, and a tentative hypothesis is formulated and explored. Finally, conclusions and new theories or practises are conducted from the results, found when evaluating and testing the hypothesis.

In this dissertation, the research question stated in chapter 1.2 is approached first in smaller fragments. This dissertation consists of five separate research targets, described here as problem cases. The steps of inductive research are repeated for four of the problem cases. The created conclusions consist of the results of these problem cases, and are used as an input for the fifth problem case, which combines the results of the earlier cases, and extends them to create a conclusion and theory to a larger problem. The conclusions of the fifth problem case provide solutions to the research question of this dissertation.

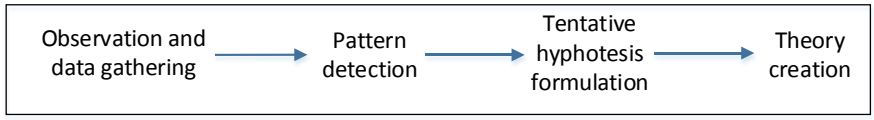

Figure 5. Inductive research steps, modified from (Blackstone, 2012).

3.2 Research method

The constructive research paradigm (Järvinen, 2004) followed in this dissertation solves several related knowledge problems, concerning feasibility, improvement and novelty (Crnkovic, 2010). The constructive research method implies building of an artefact (practical, theoretical or both) that solves a domain-specific problem, to create knowledge about how the problem can be solved (or understood, explained or modelled) in principle (Crnkovic, 2010). A construct can be a new method, model, process, software, framework or a concept. Constructive research produces results which can have both practical and theoretical relevance. Theoretical relevance provides new theoretical knowledge that needs scientific acceptance. The practical relevance refers to empirical knowledge creation that offers final benefits. The common evaluation criteria for constructs consist of completeness, simplicity, elegance, understandability and ease of use (March and Smith, 1995).

The constructive research method resembles closely the design science paradigm (Hevner *et al.*, 2004). In both approaches, the research must produce an artefact in the form of a construct, a model, a method, or an instantiation, and the produced artefact must be strictly evaluated in its intended context. However, the design science paradigm is mostly applied in the context of information systems, aiming to develop technology-based solutions (Hevner *et al.*, 2004; Peffers *et al.*, 2007; Purao, Rossi and Sein, 2010). In this dissertation, the research is performed in the context of software engineering, and most of the constructed artefacts are not technology-based. It is still worthwhile to note that in this research the information intensive systems are related to open data and data quality management. According to Purao, Rossi and Sein (2010), new integrated research approaches can be outlined that exploit the strengths of two separate research approaches. Thus, the research described in this dissertation can be positioned in between constructive research and design science.

Constructive research commonly consists of the following steps (Oyegoke, 2011): 1) Selecting a practically relevant problem; 2) obtaining a comprehensive understanding of the study area; 3) designing one or more applicable solutions to the problem; 4) demonstrating the solution's feasibility; 5) linking the results back to the theory and demonstrating their practical contribution; and 6) examining the generalisability of the results.

Figure 6 describes the features of the constructive research (Oyegoke, 2011) that is applied in the research method of this dissertation together with these common steps.

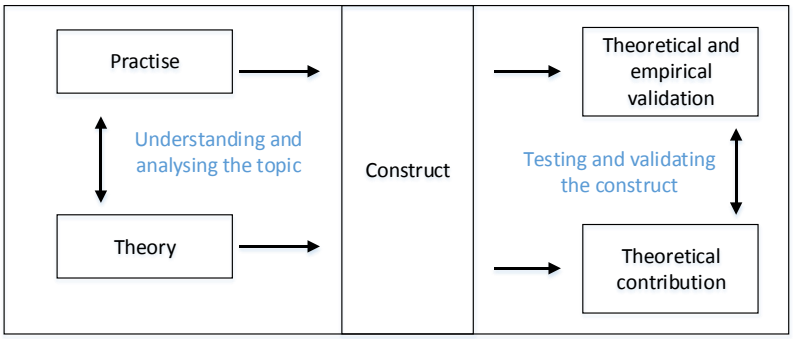

Figure 6. The features of the constructive research approach modified from (Oyegoke, 2011).

The constructive research approach begins with identifying relevant problems that have research potential. Constructive research problems are commonly approached based on anecdotal evidence, practical experience or theoretical work (Oyegoke, 2011). The purpose is to define the knowledge gap in state-of-the-art literature review, or in practice, through which the research problem can be specified. The research problem should provide opportunities to produce constructs that extend both practical and theoretical knowledge (Lehtiranta *et al.*, 2015). In this research, there were several problem cases that involved digital service and open data quality. Relevant problem cases were selected due to their different viewpoints on open data and digital service engineering. The problems could be identified both from the literature, and from practice, with the help of domain actors in all of the problem cases. Additionally, the selected problem cases all enabled inspection of the quality of data or service in different contexts, and application domains.

After the problem identification, a theoretical understanding of the topic is carried out with the help of a literature review, which is then extended to practical experience, to have a more comprehensive understanding of the problem area. In this research, understanding of the study area was achieved separately in each problem case, by interviewing the relevant stakeholders in the problem domain, and by exploring the current literature. The literature review was extensive, considering several relevant topics around the quality of services, quality of data, business, data and service ecosystems, service engineering and quality evaluation methods, and approaches for services and data. The interviews made it possible to achieve empirical data that provided different viewpoints to the same problems. The interviews were also beneficial in the cases when the subject was new and adequate information could not be found from the literature.

After the data collection, the construct is designed based on the interpretation and analysis of the literature review, and the empirical experiences from practice. In this research, solutions were designed for each problem case separately, based on the theoretical and/or empirical analysis. The constructs varied from methods to concepts, having their own approach in research problems. The methods were

described as processes, including the description of phases, activities and steps, whereas the concepts were described as a specification of concept elements.

After the design of the construct, the next step is to proof the workability of the new construct. This is achieved by testing or validation of the construct, following the principle of triangulation (Oyegoke, 2011), for example, with the help of theoretical or empirical validation, or quantitative or qualitative validation, or both. However, the validity of constructive research depends on the validity of the testing procedure of the construction, because valid results are only produced by valid procedures (Pekuri, 2013). Therefore, every method of testing the functionality of the construction must be evaluated within that context in which it is used. In this research, the demonstration of the practical contribution of each solution was implemented by the actual usage of the solution by the domain actors, testing the solution in selected case examples, using the solution in real industry case or by commenting and refining the solution by the domain experts of the industry. Thus, the validation was empirical and qualitative. The validation made it possible to find development targets, and to refine the constructs based on achieved experiments.

Constructive research demands that the construct should add to the body of knowledge. Thus, theoretical contributions should be posited; their novelty and scope of application should be clearly stated (Oyegoke, 2011). According to Lehtiranta *et al.* (2015), a failed construct produced with constructive research does not necessarily mean that the research is invalid. In any case, the results should be linked back to theory. In this research, the feasibility of the solutions was demonstrated by specifying the phases, steps or guidelines on how to use or implement the solutions. Whole new theories were created, and they were documented and compared with existing ones.

In the last step, the generalisability of the construction to the wider problem type should be assessed. In other words, the researcher should be able to assess the generalisability of the research findings. The developed construction should provide a solution to a whole problem type, not just to an individual case (Olkkonen, 1994). According to Lee and Baskerville (2003), generalising can occur in four ways: from empirical statements to other empirical statements, from empirical statements to theoretical statements, from theoretical statements to empirical statements, and from theoretical statements to other theoretical statements. In this research, the generalisability was performed both from theoretical to other theoretical statements in the case of developed methods (e.g. the QRF and RE methods, and a solution for quality evaluation of open data), and then from theoretical to empirical statement when applying the methods in experiments in different application domains and projects with guidelines and descriptions of processes and activities. Furthermore, the generalisability was performed both from the empirical (i.e. industry interviews) and theoretical statements (e.g. open data based business ecosystem) to the theoretical statements in the case of the specified concepts. The generalisability of the results was assessed in each problem case; the solutions were not targeted to any application domain, but were generic, and could be applied to any domain.

The detailed description of the research process adhered to in this dissertation is provided in Chapter 3.4, after the introduction of the problem cases.

3.3 Description of the selected problem cases A–E

This research includes five international problem cases. Each problem case is a representation of certain identified problem areas or targets, identified based on the literature and/or the actors in the domain to which this research is meant to find a solution. The problem cases are named from A to E, and are described in the following sub-chapters.

3.3.1 A: Quality specification and evaluation in software product lines

As software systems are increasingly entering consumers' everyday life, they must demonstrate high reliability and availability in order to satisfy the consumer needs. This means that they must function correctly, and without interruption. This is particularly important in the context of product lines, where faults and bad design decisions affect all the members of the product line. Therefore, an effort should be made already at the design phase to verify that the quality requirements are identified, transformed into architectural design and evaluated. Several quality evaluation methods already exist, but they embody several shortcomings. Furthermore, no methods exist that make it possible to identify and specify the quality requirements from the relevant stakeholders, and rationalise how to bring them architectural decisions. The application domains where the research was conducted were the product lines of distributed embedded systems, and information systems. The actors in the domain, such as business managers, product line owners, product line architects, domain experts and managers of reusable assets, needed a solution to the problem of how to capture quality requirements to product line architecture and products.

3.3.2 B: Requirements engineering in digital service ecosystems

When acting in a large digital service ecosystem, the interoperation among ecosystem members must be fluent in order to enable the service co-innovation and co-creation. The interoperability rules and common regulations, and the service development models and methods must be convergent to enable communication in the ecosystem, and the interoperability of the services in the ecosystem. The application domains to which this research was focused were an ecosystem of cloud services that produces content for multimedia services, and an open service platform of multi-modal mobility services. The actors in the domains included digital service ecosystem actors, such as service providers, data providers and data brokers, cloud IaaS and PaaS providers, platform providers, application developers, and service brokers. The ecosystem members were co-creating services, but no service RE method could be found that was applicable to the ecosystem con-

text. The main identified research problem was how to engineer service requirements and how to co-innovate and co-develop services in a digital service ecosystem. This included the additional problem of what kind of support the ecosystem must provide for service requirements engineering.

3.3.3 C: Requirements of open data based business ecosystems

Open data is considered interesting, both in business decision making and when utilised in digital services. However, the ecosystem context where the actors of open data business can cooperate is not yet specified. Therefore, the specification of this kind of ecosystem must consider the characteristics of open data value networks and business ecosystems. The application domains of the research in open data based business included environment monitoring, weather observation, healthcare, media, transport, UI design, mobile services, business-critical IT, and general data based services. The actors in these domains included data providers, application developers, tool providers, application users, and technology providers, who were interested in acting in open data based business and in an open data ecosystem. The obstacles for open data based business included the lack of applicable business models for open data, and the lack of supporting elements (services, models) that support open data based business. The main objective for this research was to identify the requirements of an ecosystem where open data could be provided and utilised.

3.3.4 D: Quality of social media data in service architectures

As more and more freely accessible open data becomes available, the data-users need to ensure the quality and trustworthiness of data in order to be able to trust it in their businesses. The usage of unreliable data may lead to poor or incorrect business decisions, and cause a lot of unnecessary effort and expenses for companies. The trustworthiness of data is achieved by ensuring the reliability of the data and the data source, and confirming that the quality and relevancy of data are appropriate for the specific situation of the consumer. The importance of data quality evaluation has commonly been recognised, but no relevant progress has occurred to rationalise and standardise data quality evaluation. The application domain of the research on data quality evaluation was decision support systems in business operation, in big data consulting, among big data architectures. The actors in the domain were the company's decision makers that were interested in making business decisions based on open data. A lot of data from social media was available for the decision makers, but its trustworthiness and value was unknown. Several relevant problems could occur in open data utilisation in business, most importantly; how to find trustworthy open data, how to evaluate the quality of the open data, and how to find out the value of the open data in decision making.

3.3.5 E: Quality management in open data based digital service ecosystems

When engineering digital services that utilise open data, the quality of the data must be ensured. In the context of digital service ecosystems, poor and unknown quality of data causes larger scale problems, affecting all the ecosystem services, and thus the whole trustworthiness of the ecosystem. Therefore, the ecosystem must provide the means to certify the quality of open data. The elements of open data based business ecosystem, the solution of open data quality evaluation and the ecosystem based service engineering approach must all be combined under the same ecosystem context. The different actors of open data and digital services must find their own role and place in the ecosystem. The domain of this research was the open data based service ecosystem applied to digital healthcare, multimedia, transportation, and business decision making (in service creation and operation). The actors in the domain, i.e. the digital service ecosystem actors and open data based ecosystem actors, faced two main problems; how to engineer services in an open data based ecosystem, and how to ensure the quality of open data utilised in services.

3.4 Research process and results

In this dissertation, the research is conducted in three main process steps (Figure 7) that cover the common steps of the constructive research method (Oyegoke, 2011). The steps are used separately for each of the problem cases:

1. Collection and analyses of data by conducting a state of the art review of the current literature and the state-of-the-practise review among field representatives in the selected problem area.
2. Developing method/process/concept constructs, i.e. producing the constructs as an output of the analysis.
3. Evaluating method/process/concept constructs in the selected problem areas and reflecting the findings back to the theory and practice of the problem areas.

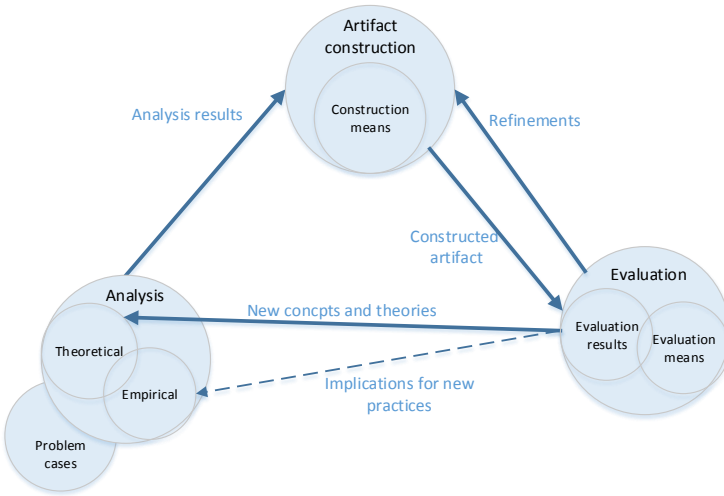

Figure 7. Description of the research process.

3.4.1 Analysis: theoretical & empirical

Problem case A: The starting point for the research of software quality evaluation was the lack of methods for identifying quality requirements, transforming them to architectural models, evaluating whether the requirements were met, and tracing the requirements to design decisions, and vice versa. Therefore, the theoretical research was implemented with the help of a literature survey of existing R&A prediction methods, applicable at the architectural level, utilising the results of the methods in the previously conducted survey (Immonen and Niemelä, 2008). The survey represented a comparison framework that defined the required characteristics of the analysis methods from the context, user, method content and evaluation perspectives. Some frameworks and methods exist that capture quality requirements, and methods for modelling family architecture. However, a gap was identified between the quality requirements and the quality modelling in the architecture. Therefore, this research concentrated on filling this gap with the help of literature from both requirements engineering and architectural modelling.

There are multiple stakeholders involved in software system or service engineering, with different goals, especially in the case of product lines. At first, the requirement sources must be identified. These were defined to include a) the markets (customers, end-users, etc.), b) business (marketing managers, product line owners, etc.), c) product line (product line architects, manager of reusable assets, domain experts, etc.), d) product (product architects, developers, maintainers and other development staff), and e) other (developers of services/products/applications that use the product/a part of the product, etc.). Next, the theoretical research concentrated on requirements engineering methods that consider quality requirements. Some of them provided ideas that could be applied,

such as in representing variability between the members of the product line, and in trade-off analysis. Finally, different solutions to extend or annotate the architectural notations, to represent the quality requirements in the architecture were examined. The separation was made between the required and provided quality; the required quality guides the design of concrete architecture, and helps to make design decisions, whereas the provided quality represents the implemented quality that can be used, for example, in quality analysis.

Problem case B: When starting to explore the literature about digital service ecosystems, initially no work on digital service engineering in the ecosystem context could be found. Furthermore, the formal definition of digital service ecosystems could not be found from the literature. Therefore, a theoretical research was conducted with the help of a literature review, consisting of related topics; service co-innovation, service value co-creation, enabling infrastructure and utilisation of the ecosystem's assets. Although some of these topics were quite mature, they were not introduced in the service ecosystem context. However, they all embody a part of the idea of ecosystem based service engineering, and should therefore be represented in the ecosystem. Open innovation enables the co-creation of ideas for a service with other actors of other ecosystems, whereas the value networks formed by the ecosystem members enable the co-creation of value inside the ecosystem, achieving the common goals together. Ecosystem infrastructure enables the collaboration and co-operation of ecosystem members, and the existing ecosystem assets provide the assets for service engineering, and acting in an ecosystem. In addition, since the concept of digital service ecosystems itself was not properly defined, the literature research also concentrated on the properties of the business ecosystem, digital service ecosystem and software ecosystem, and made comparative definitions. The term digital service ecosystem was specified with the help of a definition of the digital ecosystem (Chang and West, 2006) and the definition of service ecosystems (Liu and Nie, 2009; Riedl et al., 2009; Ruokolainen, 2013).

Furthermore, the content of the elements of the ecosystem; members, capability, infrastructure and existing assets, had to be specified from the digital service engineering viewpoint.

Problem case C: At the time when the first research on open data was conducted, there was no clear and unambiguous definition for open data based business ecosystem. Therefore, the theoretical part of the research was implemented with the literature review, considering data value chains (Chen *et al.*, 2011; Kuk and Davies, 2011; Poikola, Kola and Hintikka, 2011; Tammisto and Lindman, 2011), business models of data (Osterwalder, Parent and Pigneur, 2004; Baden-Fuller and Morgan, 2010; Perr, Appleyard and Sullivan, 2010; Teece, 2010; Tammisto and Lindman, 2011) and open communities[11]. All these topics reflected open data based business ecosystems from the different viewpoints. The empirical part of this research was implemented among business stakeholders of open data, with the intention to collect requirements for the open data based business

[11] http://open-data.europa.eu/en

ecosystem. The representatives of 11 Finnish companies were interviewed, including ecosystem actors such as data providers, application developers, infrastructure providers and application users. Companies were selected from different application domains to the interviews. The interviewees, for example product developer managers, customer and development managers, and finance and administration managers, were selected based on their knowledge of the business viewpoint of their company. The interviewed companies differed according to the company size, application domain and service types. Occasionally companies had more than one role in the ecosystem.

The interview consisted of two parts. The first part consisted of a semi-structured interview (Järvinen, 2004) that enabled variability in conversations, due to the different background and application domains of the interviewees. The interview consisted of general questions about the usage of data and open data in a company, the benefits and challenges of open data, and the business potential that the open data provides. The second part of the interview was implemented as an open-ended theme interview (Hirsjärvi and Hurme, 2001; Livesey, 2007), where the interviewees were asked to freely discuss the four main themes; open data, applications, co-creation and open data ecosystem based business, and several sub-themes related to these main themes. Each interviewee selected a role, which represented his/her company's role in the ecosystem. The roles were defined based on the literature survey analysis. The interviewee inspected the themes from the viewpoint of this role. After the interviews, a comprehensive analysis of the results was conducted. The analysis made it possible to identify the challenges and opportunities of the open data, applications and services of open data, and to evaluate the feasibility of the open data ecosystem, and most importantly, to identify the requirements of the open data based business ecosystem from the viewpoints of different actors.

Problem case D: The identified challenges in open data and big data quality formed the starting point for the research on open data quality evaluation. Interviews were performed with two companies that utilised or were willing to utilise open data in their business. These companies had realised that free-formed discussions, for example, in social media, can provide insight into consumers' opinions, preferences and requirements with regard to the company or its products/services. Therefore, the companies were willing to invest in research on the utilisation and management of open data in business. However, the companies did not want all the data that was available, since in the case of big data there would be huge amounts of data to be processed. The companies wanted to filter the data and take only that which they see as reliable and valuable for them. The purpose of the interviews was to identify the challenges with regard to open data in actual business usage, and to outline a solution to describe the new data related business case of the company. The interviews with the company's decision makers were performed as a semi-structured interview, with the general questions. The interviewees were first asked to describe their business case, where the data would be utilised, after which the current status of the data usage in the company was surveyed. The questions of the interview were divided into business, func-

tionality, data, constraints and non-functional requirements. The business related questions were specified according to the business elements of the Business Model Canvas (Osterwalder, Parent and Pigneur, 2004), which assisted in outlining the business elements, after which the requirements considering the functionality, data, constraints and quality of the solution were outlined. The interviews with the company decision makers revealed that the gathered data of the companies come from the different levels of sources; there is no evidence about the quality of the data. Furthermore, there is a lack of applicable methodology for data users to verify the quality of open data in their business processes.

Literature research started after the analysis of the interviews, with the analysis of data quality attributes and metrics, and metadata standards. The existing evaluation methods were surveyed, and their deficiencies were identified. The literature research continued to big data architecture, on how to manage data through the organisation's processes, and how to manage data with the help of data quality policies. The aim was to find a solution of how to evaluate the quality of data, and manage the quality of data through the company's business processes.

Problem case E: The starting point for the research on open data based digital service ecosystems were the results of problem cases A, B, C and D. The problem cases A and B enabled the description of the service requirements engineering method, and the problem case B provided the environment; the digital service ecosystem with its required elements. The problem case C provided the elements of the open data based business ecosystem that are necessary for the digital service ecosystem. The problem case D presented the solution for open data evaluation.

In problem case E, the existing ecosystem elements from the problem cases A to D; the knowledge management models and support services, were defined in more detail to enable the usage of open data of which quality had been certified. This was achieved by identifying the required action of the capability model to enable the quality certification, which was then supported by the knowledge management models and support services. The term open data service was specified to mean that the open data is brought into the ecosystem as a service that encapsulates the data for the usage of digital service developers. Furthermore, the need for a new ecosystem element was detected; the core was specified, that acts as an integration framework for combining the knowledge models, and support services for developing open data services and digital services based on them. Although several application domains were presented, such as healthcare, multimedia, and transportation, the research was conducted at a more generic level in order to be applicable to any domain.

3.4.2 Construction

Problem case A: The literature survey of the existing methods and approaches for reliability prediction and evaluation at the architectural level (Immonen and Niemelä, 2008) enabled detection of the common features of the methods, but also specified the required features for a method for identifying reliability require-

ments, transferring them to architecture, and evaluating whether the requirements are met. With the help of the literature analysis, the method for capturing quality requirements (QRF method) was developed. The first step of the methods supports the identification and definition of the quality goals, and a means of mapping the quality goals to the stakeholders. In the second step, the most important quality requirements and their interest groups are represented, scoping the quality requirement to architecture, components and applications, and defining how the quality requirements relate to business and other capabilities of the product line. In the third step, the quality variability and points of time when variation takes place are identified. In the fourth step, standard service taxonomy, explicit mapping of quality requirements to services, and domain based clustering are presented. Finally, the qualities are represented in architectural models using the quality profiles that consist of quality dimensions and values. The steps of the QRF method were clearly described with the purpose, the main activities, and contributions of each step. The use of the method was exemplified with the laboratory case; the Distribution Service Platform (DiSeP). Furthermore, the guidelines for applying QRF were described in order to assist the product line stakeholders to use the method.

Problem case B: The design of the ecosystem elements and service engineering model was implemented based on the requirements arising from the literature survey. The concept of digital service ecosystem was specified with the help of the definition of the ecosystem elements, i.e. actors, capability model, infrastructure the digital services, refining the existing definitions of these elements with ecosystem based service engineering related properties, responsibilities and requirements. Ecosystem actors and actor roles were defined to distinguish the roles and responsibilities of each actor in the ecosystem. The capability model was defined to describe the properties of the ecosystem, and how these are implemented using the ecosystem services that the ecosystem infrastructure provides. The infrastructure was specified to provide models and assets that assist in the RE, including domain model, service engineering model, knowledge management models and ecosystem support services. The digital services are the main result of the ecosystem, and the RE process can either result in new digital services, or they are mapped to existing digital services. The requirements can also be identified as new ecosystem support services, or they can cause changes in existing ones.

The ecosystem based service engineering model was specified to consist of the five phases, using which the service requirements are engineered, modelled and validated. A more detailed specification was created for the first three phases in the form of a scenario-based service RE method. The RE method was developed using the features of the QRF method as a starting point. A scenario-based technique was selected to engineer requirements, as it could describe both the viewpoints of RE: business and usage. The process description of the method usage was specified to describe the activities for RE in the ecosystem. Two templates were developed to support the activities of the RE process: The Use Case Description template for service innovation, and the Use Case Analysis template for

assisting in identifying, analysing and specifying requirements. These templates, as well as the templates for business requirements estimation and the QA specification template, were meant to be provided by the ecosystem's knowledge base. Furthermore, the service engineering method, the RE method, and the RE process description were included in the ecosystem's knowledge models.

Problem case C: The initial outline for the open data based business ecosystem was specified according to the knowledge analysis from the literature. This included the classification of ecosystem actors, their role in the ecosystem, and the possible business potential of each actor. The initial outline was refined based on the interviews among industry. For the interviews, a method for systematic data and information collection, analysis and presentation was specified to be applicable to the different ecosystem actors. Company representatives responded to the questions in the role they had selected from the outline, describing their motives, risks and requirements in acting in an open data ecosystem. Based on the analysis of the interviews, the viewpoints of the different actors in different domains (e.g. environment monitoring, healthcare, media and transport) were combined at a generic level. The analysis resulted in the concept of an open data based business ecosystem. Several new actors and their roles were identified with the help of the interviews, and new support services were identified to support the businesses of the different actors. The created ecosystem concept specified the ecosystem actors, and the capability of the ecosystem to support business model elements of its actors, including the necessary supporting services, and support for business model elements.

Problem case D: The analysis of the literature research made it possible to outline the data quality evaluation phases and targets for the solution. By integrating these to the requirements arising from the actors in the domain, a solution was developed to evaluate the quality and trustworthiness of open data. The solution was first outlined as use case diagram that described the interaction between the user and the solution, after which the more detailed data models were described. The solution was designed and implemented as a big data service architecture, and was co-developed together with one of the companies that participated in the interviews. The company focused on big data use case R&D. The REST application programming interface facilitated independent work on activities of the organisations, and the agreement of a common integration interface.

The solution helped conduct the data quality evaluation in several data processing phases of the big data service architecture, going through the pipeline of the big data system, and finally providing the valuable, analysed data to the company's business decision makers. The solution specified the main elements and phases of the evaluation (i.e. architectural elements and quality policies with metrics and algorithms used) and the evaluation process, with supporting tooling. The solution also defined the data quality attributes and metrics, and their applicability to different cases, thus being applicable in different contexts and situations. Data quality evaluation and the management of data quality are controlled by data quality policies, which each organisation (i.e. ecosystem member) can specify to fit their own purposes and situations.

Problem case E: The earlier definitions; the concepts of open data based business ecosystem, the ecosystem based service engineering model, and the data quality evaluation approach, were brought together and refined, and the Evolvable Open Data based digital service Ecosystem (EODE) concept was specified. The EODE concept (Figure 8) includes the following:

1. *The ecosystem's capability model,* which describes quality related activities for governance and regulation actions of the ecosystem, for open data certification, and for service engineering related actions.
2. *The knowledge management models,* which describe the common knowledge of the ecosystem, comprising generic and domain-specific parts.
   Generic models:
   - *Ontologies* that conceptualise elements related to data, quality, metrics and services. For example, a reliability ontology or service ontology.
   - *Design time artefacts:* i.e. architectural styles and patterns. Ecosystem members also share the integration architecture represented as a common knowledge model. Other common knowledge models may include service description ontologies, service component models, quality of service models, service composition models, and service community models.
   - *Quality policies* used in quality evaluation and management. These include a) the ecosystem policy that defines a set of governance services common to all ecosystem members, and rules on how to configure and monitor these services, b) filtering policy that defines what open data sources are acceptable in the ecosystem, c) quality evaluation policy that defines the quality attributes, metrics and rules for their applicability to quality evaluation, and d) decision making policy, that defines how the decisions are made based on the strategic or tactical operations of the ecosystem.
   - *Service engineering models,* that are equipped with the methodology and tools for developing open data services and digital services.
   - *EODE service model*, which is a generic service model for all kinds of digital services.

   Domain-specific models:
   - *Domain models* that define the domain specific quality attributes, variations between the domain and the common knowledge management models, and the adaptation rules for mapping variable elements in the context of the EODE.
3. *Support services* that implement the activities of the capability model, including the taxonomy of support services for data quality specification, evaluation and management.
4. *EODE core*, that is the integration framework that registers and manages the open data services, digital services and support services, and also provides knowledge, engineering and domain models as services, containing the mechanisms to control the ecosystem.

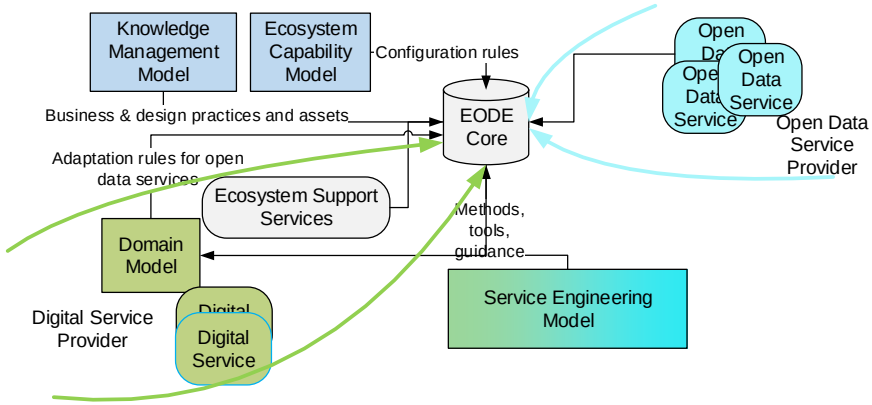

Figure 8. Overview of the EODE concept (presented in Publication V).

The data quality certification process was also specified for the ecosystem that certifies the data, utilising these knowledge management models and support services. The process makes it possible to bring open data to the ecosystem, transform it to a usable form for the ecosystem, validate it against its intended usage, monitor the data sources and the usage of the data, and continuously evaluate the quantified value of the open data service, thus certificating the quality of the data for the ecosystem and its members.

3.4.3 Evaluation

Problem case A: The evaluation of the developed QRF method consisted of the trial usage of the method in different kinds of application domains and product families. The method was applied to four case studies, where it was validated in connection with the Integrability and Extensibility Evaluation (IEE) method (Henttonen *et al.*, 2007) and the Reliability and Availability Prediction (RAP) method (Immonen, 2006). The IEE method (Henttonen *et al.*, 2007) enables extensibility and integrability evaluation from software architectural models. The IEE method was applied to a case study of an open source tooling environment for software architects and designers called Stylebase for Eclipse. The QRF method covered the first phase of the IEE method, including the activities of the impact analysis, quality analysis, variability analysis and hierarchical domain analysis, and resulted in a list of prioritised quality criteria, against which the architecture was evaluated.

As a part of the RAP method (Immonen, 2006), the QRF method was used in three cases; to validate the DiSep platform (Niemelä *et al.*, 2005; Immonen, 2006), in the Personal Information Repository (PIR) (Immonen and Evesti, 2008), and in the development of a SMEPP middleware in connection with the quality aware architecting approach (Ovaska *et al.*, 2010). DiSep (Matinlassi, Niemelä and

Dobrica, 2002) was a distribution platform for a system family of software systems that is formed by executing units in a networked environment. PIR (Lähteenmäki, Leppänen and Kaijanranta, 2008) was a reliable business-to-consumer (B2C) document delivery system between customers and service providers. SMEPP was a secure middleware for embedded peer-to-peer systems. The QRF method supported the first phase of the RAP method, enabling the separation of family and system-specific requirements, and describing variability between family members. It also assisted in mapping family-specific requirements to the family architecture, and the system-specific requirements to the system architecture. The case studies showed that the QRF method is able to work without any special architectural models. Some minor modifications to the method were also implemented, and further development targets were identified. After the refinement, the guidelines for applying the QRF method were provided and delivered in international audiences as a tutorial, and also stored in a methods repository.

Problem case B: The RE method was validated in two international projects that acted as digital service ecosystems, and used the developed RE method. The RE method was first applied in the ITEA2-ICARE project, in which the digital service ecosystem included 25 service ecosystem members from five countries in Europe, providing and using digital cloud-based services related to the operation of end-to-end interactive multi-screen TV services. The goal for applying the RE method was to collect and analyse requirements from the ecosystem members towards a shared service-oriented platform, enabling the provisioning, integration and use of services amongst the members of the ecosystem. The RE method was also applied in the CDC project, which aims to develop an open service platform, offering open real-time data from several data providers (offering data normalisation, integration and analysis, service hosting, open data APIs, service registries and platform modules and services to third-party application developers). The project included seven partners from four European countries. The goal of the RE method application was to extract high-level user and business requirements for the open real-time data platform to be developed. All in all, valid results were achieved with the help of the RE method; altogether nearly 275 requirements were identified in the ICARE project, and 23 in the CDC project, including functional, non-functional and business requirements, and constraints.

After the usage of the RE method, feedback collection was performed among the partners that were involved in the requirement engineering. The purpose was to receive user experiences and opinions about the method, and to find out its advantages, shortcomings and development targets. The feedback collection was implemented using a web-based questionnaire (Davis, 1999) that was accessible through a web page to the project partners that filled the Use Case Description and the Use Case Analysis templates. According to the feedback analysis, the service RE method was seen as valuable and useful in the beginning of the service engineering process, when starting the long-term development of new service architecture for digital ecosystem-based services, and in describing, documenting and communicating the capabilities of the digital services and the service architecture they require. The method was also seen as useful in the analysis phase,

where the different stakeholders work together. However, the definition of quality requirements was identified as the development target. The knowledge management model of the ecosystem is responsible for providing the ontologies and the methods to be used in each RE phase to achieve the non-functional requirements. Overall, the templates were good for documentation and communication purposes, especially in a large European project. Furthermore, the enquiry template and feedback collection process with tooling resulted in fast feedback inside the ecosystems.

Problem case C: The proposed ecosystem concept was evaluated by the same industry representatives that participated in the interviews. First, the textual concept description was sent to the interviewees via e-mail, and the interviewees were asked to comment and suggest improvements to the concept. The feedback could be provided in a textual format via an electronic version of the concept description. Only minor modifications were identified, based on which the concept was refined. The final concept was communicated to the industrial representatives in workshops, in which a summary of the interviews was represented. The workshops were also used as a means to collect final feedback. Representatives from the interviewed companies were formally invited to the seminars in Oulu and Espoo. The representation in the seminar included first the results of the general interviews, and then the results of the theme interviews, based on which the ecosystem concept was created. The representatives were asked to comment on each topic. The representation generated discussions about the most vital issues for the ecosystem, and made it possible to weight the most important topics of the findings.

Problem case D: The solution for data quality evaluation and management in a company's business process was validated in a case example with a case company. As a starting point, the company provided metadata information of extracted Twitter data sets, which is utilised as a basis for sentiment analysis. The company used the proposed solution to find interesting data from Twitter. The main purpose of the company was eventually to combine social media data with its own internal data to achieve customer insights that could be utilised in business decision making. The quality policies had a great importance in quality evaluation; the definition of these policies is an organisational issue, and is a required prerequisite for using the solution. By pre-defining the organisational policy, the case company could select relevant data sources. The attributes and metrics were selected automatically based on selected data sources. The evaluation itself was also automated. By pre-defining the decision-making policy, the case company could select the relevant data for decision making, weight the relevant quality attributes, and define adequate values for the quality attributes case-specifically. In the decision-making process, the relevant data was visualised to the end-user according to the decision making policy. By configuring the policy, the data sets with lower quality values could be visualised, or the weighting of the data sets could be changed. The development of the solution enabled more detailed definition of roles and responsibilities of business decision makers, and provided the experience of co-development of the solution with an industrial partner.

Problem case E: The development of the EODE concept was carried out incrementally in several international and national research projects. Therefore, the evaluation of the building blocks of the EODE was implemented in a cross-domain evaluation. The QRF method was validated in distributed embedded systems, and information systems domains, and when the context changed from the product lines to digital service ecosystems, the resultant RE method was validated in a multimedia domain, and a multi-modal mobility services domain. The generic concept of an open data based business ecosystem was validated in the domains of environmental monitoring, weather observation, healthcare, media, transport, UI design, mobile services, business-critical IT, and data based services. The solution for open data quality evaluation was tested and evaluated in a data consulting domain. The evaluation of these building blocks was described in the problem cases A, B, C and D, and their transformation to the EODE context was described in this problem case. The infrastructure of the EODE was also described in the problem cases B and C, and further refined in this problem case. As the integration framework, the Digital Services Hub was used to register and monitor any kind of digital entities that had a digital API. The Digital Services Hub was developed in the scope of the ITEA2-ICARE project in 2014–2015. The semantic data model, which enabled the more intelligent service discovery and intelligent service matching, supporting also interoperability between services, was developed in the Digital Health Revolution (DHR) project in 2015. The Digital Services Hub was evaluated in the ITEA2-ICARE project, where 25 ecosystem members from five European counties registered their services, and used the Digital Services Hub for authorising and visualising service connections. The Digital Services Hub fulfilled its purposes well in the multimedia domain of eight international service providers. However, the context was closed, and therefore more validation is required.

The validation of the whole EODE concept was performed at the conceptual level with the help of a five-phased data quality certification process (Figure 9). The process consisted of the following steps that validated the quality of the open data in the ecosystem context:

1. The relevant open data sources were searched, and the evaluated and accepted data was extracted from these sources.
2. The syntax and semantics of open data was checked and transformed to a standard format
3. The quality evaluation of open data services was implemented by the data consumers (i.e. the digital service providers).
4. The quality policies were changed, based on changes in open data sources and/or (quality of) open data
5. The open data services for the ecosystem were valued.

The developed process description specified the support services, and the knowledge assets (i.e. the quality policies) required to implement each process activity. Furthermore, it described how each activity was performed, how the content of each quality policy was used in different activities, and how the quality policies were described and configured. The application of the concept to the

different application domains and business fields is naturally the next step in the concept validation.

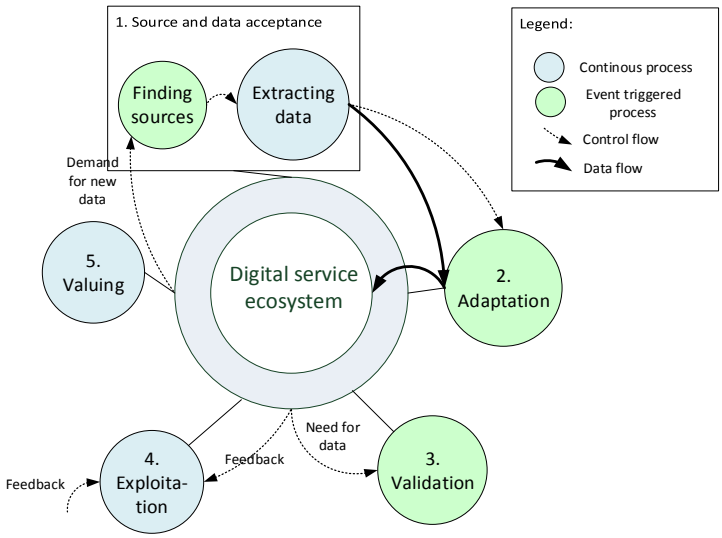

Figure 9. The phases of the open data quality certification process (represented in Publication V).

3.5 Summary of research activities

Table 5 summarises the research activities of each problem case. The table maps the related features of open data based digital service ecosystems (Table 4) to the problem cases. In addition, the table introduces the produced new constructs that support the related features of open data based digital service ecosystems, and describes how these features have been evaluated. The cross-domain evaluation is described more thoroughly in Figure 10.

Table 5. Summary of the research activities.

Problem cases	Related features	Collected data items	Produced constructs	Evaluation
A: Quality specification and evaluation in software product lines	2. Service co-development 3. Knowledge-based service engineering	Methods for requirements engineering and architectural modelling R&A prediction methods applicable at the architectural level	A method for capturing quality requirements (QRF method) Guidelines for applying QRF	Applying the QRF method in four case studies, and modifying the method according to the observations
B: Requirements engineering in digital service ecosystems	1. Service co-innovation 2. Service co-development 3. Knowledge-based service engineering 4. Enabling infrastructure	Methods for service requirements engineering, innovation, and co-development Methods for knowledge based requirements engineering	Concept of a digital service ecosystem Service RE method, including RE process description and templates to support the activities of the RE process	Applying the RE method in two ecosystems. Feedback collection among partners that were involved in RE in ecosystems
C: Requirements of open data ecosystems	4. Enabling infrastructure 5. Open business model	Actors of business ecosystems, data value chains and business models of data Requirements for the ecosystem from industry	A concept of an open data based business ecosystem	Evaluation of the concept by the same industrial representatives that participated in the interviews
D: Quality of social media data in service architectures	6. Quality evaluation of open data	Challenges with regard to open data in actual usage in industry Data quality attributes, metrics, metadata standards. Data quality evaluation methods and big data architectures	A solution for quality evaluation of open data in service architectures	Developing and validating the solution with a case company that applied it in the case example
E: Quality management in open data based digital service ecosystems	6. Quality evaluation of open data 7. Support for cooperation of the actors of open data and digital services	Integrating the updated scientific knowledge and collected practical observations from the cases A to D	Concept of an open data based digital service ecosystem	Separate solutions validated in cases A to D A plan for evaluating the EODE concept as a whole

Figure 10 summarises the development of the EODE concept through the problem cases. The developed constructs of the problem cases A to D assist in specification of the EODE capability model and the infrastructure with support service and knowledge management models. Furthermore, the developed solution for quality evaluation of open data in the problem case D is used as a starting point for the specification of the process for data quality certification. The new elements of EODE specified in problem case E, the core[12] and the semantic data model[13], were validated in other projects. Each problem case assists in achieving understanding of the features of the ecosystem over the specific case, thus helping to achieve the features of Table 4. The evaluation of the constructs in each problem case is a starting point of the cross-domain evaluation of the EODE. More evaluation of the EODE concept is still required. The evaluation plan is described in subsection 5.6.

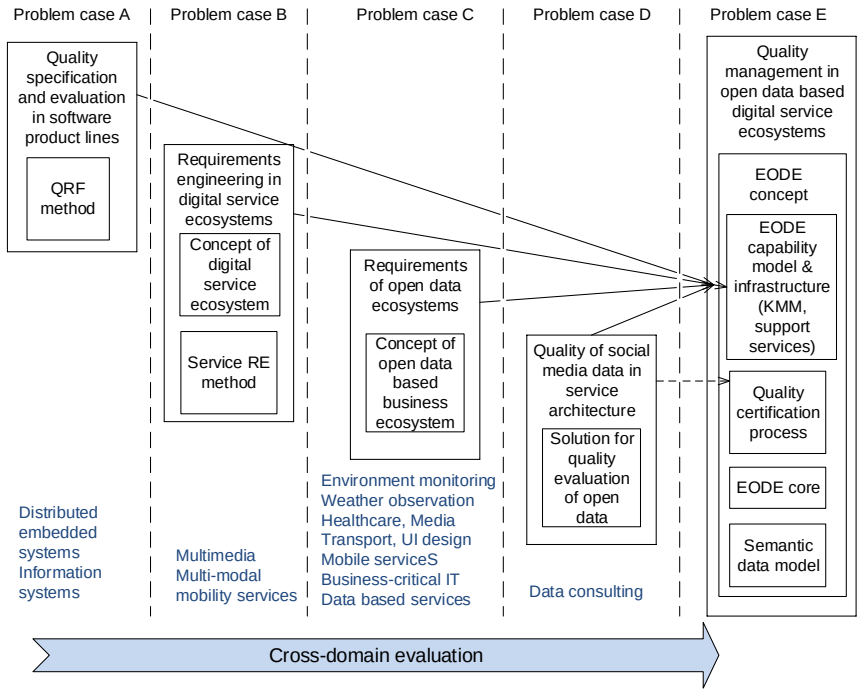

Figure 10. Development of the EODE concept.

[12] https://www.digitalserviceshub.com/registry/ (Accessed: 4 November 2016)
[13] http://www.digitalhealthrevolution.fi/ (Accessed: 4 November 2016)

4. Original publications

This dissertation consists of five original publications, which were published in scientific journals between 2007 and 2015.

Publication I represents the QRF (Quality Requirements of a software Family) method, that describes how quality requirements have to be defined, represented and transformed to architectural models. The QRF method is based on the research of Immonen and Niemelä (2008), where a novel survey of the existing R&A prediction and analyses methods was provided from the viewpoint of software architecture. The survey of Immonen and Niemelä (2008) resulted in a comparison framework, which described the required characteristic of analysis methods. In addition, the survey revealed that there were no systematic approaches available for defining and transforming quality requirements, including variable requirements, to the models of a product family architecture. Publication I used these identified characteristics as a starting point for the method development.

The QRF method consists of five steps, which enable the quality-driven software engineering, and the quality evaluation at an early phase of software engineering, in the context of product families. The first step, impact analysis, identifies external quality goals by examining how business stakeholders' needs and markets affect the scope of the family. The second step, quality analysis, separates the business, constraints and functionality related quality concerns, and expresses the quality requirements in a way that they can later be traced and measured. The third step, variability analysis, identifies the variation in qualities, and defines the variation dependency of quality requirements on business domains, and on different stakeholders. The fourth step, hierarchical domain analysis, combines the information from steps 1–3, and provides the required information for modelling and evaluating the architecture. Finally, the last step, quality representations, describes the quality requirements in architecture by using a set of views. The method has been proven to work in cases with tens of quality requirements. The QRF method is suitable for any software system where quality has a major role. Especially in the context of product families, the existence or non-existence of specific quality attributes has a broader impact on all family members.

Publication II concentrates on service engineering in digital service ecosystems. Since the concept of digital service ecosystems was relatively new at the time when this research was conducted, comparative definitions of the properties of the business ecosystem, digital service ecosystem and software ecosystem

were first specified. This publication describes the main elements of a digital service ecosystem, and their responsibilities in service engineering, and specifies an ecosystem-based digital service engineering model. The results from Publication I can be applied to the ecosystem's context, as commonalities could be identified between product families and ecosystems, for example, in both cases common knowledge is utilised. Each of the ecosystem's main elements; members, capabilities, infrastructure and the existing ecosystem assets, have their specific roles in the service engineering.

The service engineering model consists of the five phases. The service innovation phase identifies the ideas for new services, scopes and analyses them, and finally transforms them into service requirements. In the business analysis phase, the requirements that have the most business potential are identified. Business analysis also identifies how to implement the requirements. The phase for requirements analysis, negotiation and specification provides a complete requirement specification of the needed services. In the modelling phase, the requirements specification is taken as an input for mapping requirements for new or existing ecosystem services. Finally, in the validation phase, the services are evaluated and tested against the requirements. The research describes the first three phases in more detail, and specifies an ecosystem-based service RE method. The RE method was validated in two industrial cases, where the ecosystem members used the RE method for specifying digital services and related support services. The method was found to be useful for describing, documenting and communicating the capabilities of the digital services. This method was especially useful in the requirements analysis phase, where ecosystem members co-innovated and analysed the service requirements together. The service RE method introduced three main phases as a continuous and iterative engineering process that starts from business and end-user goals, and provides service taxonomy and a set of master use cases as an outcome.

Publication III describes the initial research on open data, in which the first draft of open data ecosystem from a business viewpoint is specified. The outline and the requirements of the ecosystem are collected based on novel knowledge explored from the literature, and novel practices on data based business in the industry. With the help of interviews of industrial representatives, the motives and challenges of acting in the open data ecosystems were also identified. The research identified new actors and roles for the ecosystem. For example, the data promoter actor was extended to include roles of data promoters, distributors and matchmakers. The interviews also revealed ecosystem services that were required, that must be defined and implemented while establishing the ecosystem. These included, among others, services for finding data, services and partners, services for data validation, and services for definition and standardisation of data and data interfaces. The results showed that open data based businesses can bring both direct and indirect benefits. However, there are still obstacles that complicate the utilisation of open data. One of the most significant challenges was the unknown quality of data. The interviews revealed several motives and advantages for joining the ecosystem, but also obstacles that should be carefully considered

and solved. However, according to the interviews, the level of interest in open data and open data ecosystems is high. Thus, the ecosystem could provide great benefits for the actors and their businesses through open data, and the services and applications around them.

Publication IV concentrates on the quality evaluation of open (social media) data, and represents a three-phased solution for data quality evaluation. The work specifies the elements and phases required for data quality evaluation and management in big data architecture. In the data extraction phase, the quality evaluation focuses on data provenance and data quality from the viewpoint of the situation at hand. In the data processing and analysis phases, the evaluation focuses on different quality aspects of the data. Finally, in the decision-making phase, the evaluation focuses on the trustworthiness of the data.

The data sets and metadata are managed with the help of quality policies. Organisational policy consists of the set of rules that describe what and how to evaluate to achieve data that can be trusted in a specific situation. Therefore, the organisational policy specifies the acceptable data sources, and describes the relevant quality attributes applicable to the context of the task at hand, the applicability time of the attributes, which evaluation metric should be used to evaluate each attribute, value range, acceptable value and applicable rules. The decision-making policy describes the relevant data sets for a certain situation, how to weight quality attributes depending on the relevance of the different quality attributes for the task at hand, and how to perform the decision functions.

The research classifies data to a) any freely available data (i.e. open data), b) deliberately collected external data (e.g. for market analysis and competitor analysis), c) customer feedback data, and d) company's internal data. The data is classified according to the data source type, such as social media data, feedback data, product data, competitor data, history data, or production data. This classification assists in the selection of applicable quality attributes; thus, in addition to open data, the approach can be obeyed for other types of data as well. The solution is validated with the help of an industrial case example, where it provides verified data from Twitter to help the company's business decision-making processes. The proposed solution improves business decision-making by providing real-time, validated data to the consumer. The solution may be adapted to different contexts, as it enables the data consumer to configure quality policies and apply them in a suitable way to a certain situation. The solution is also extendable – it allows inserting new data sources and data sets for data extraction, as well as new metrics and algorithms for data evaluation.

Publication V combines, adapts and extends all the earlier work (Publications I–IV), and introduces the concept of evolvable open data based digital service ecosystem, the EODE. The work extends the support services and knowledge models of the digital service ecosystem, and specifies new services and models required for open data quality certification. The support services for data quality evaluation include eight main categories; utility services, data matchmaking services, monitoring and evaluation services, recognition services, adaptation services, open data analysis services, visualisation services and tool services. The

knowledge management models include four types of quality-related knowledge; ontologies, design-time artefacts (including the service engineering model), domain models, and policies used in quality evaluation and management.

The quality policies; data filtering, data quality evaluation and decision making policies, have different purposes in each evaluation phase in the ecosystem. The filtering policy evaluates whether or not to accept the data set to the ecosystem with the help of quality metrics and rules for evaluation. The data evaluation policy is used to evaluate different kinds of data in the data extraction phase, and in data monitoring and decision making. Furthermore, each service provider specifies his own evaluation policies when searching data for a certain purpose. Finally, the decision-making policy defines the criteria for actions based on the results of the data quality evaluation.

This work also introduces a five-phased open data certification process, using which the open data is brought to the ecosystem, and certified for usage of digital service ecosystem members. In the acceptance phase, the data source, the data content, and the data quality are evaluated, and thus the open data is certified for ecosystem usage. In the adaptation phase, the open data is transformed or adapted to an open data service that can be used as a building block in digital services of the ecosystem. The third phase, validation, is performed by a digital service provider that validates the open data against its intended use in a digital service. In the exploitation phase, the open data sources and the usage of the open data are monitored. In the valuing phase, the quantified value of the open data service is continuously evaluated, and compared with other open data services, and decisions are made concerning keeping the service or substituting it with another service.

The EODE was developed based on the existing knowledge on capturing quality requirements in ecosystem based digital service engineering, and evaluating quality of open data achieved in the publications I to IV. Thus, the EODE concept was developed and validated incrementally in several international and national research projects.

5. Discussion

In this chapter, the results of the research are first discussed and their response to the research question is analysed. After that, the theoretical contributions and empirical implications of the research are evaluated. The scientific validity of the research is evaluated, and a comparison of the research to related work is performed. Finally, the limitations of the research are discussed, and future development targets are identified.

5.1 Research question and objectives revisited

This research consisted of five problem cases. The results of the first four problem cases were combined and extended in the fifth problem case to provide a solution to the research question stated in the introduction: “How to design the quality of digital services in open data and ecosystem based service engineering?”

To reach the solution to the research question, three objectives were identified in the beginning of the research. The first objective was to examine the transformation in service development when moving from the closed environment to more open ecosystems, and to specify the main elements and phases for digital service engineering in the ecosystem. The service engineering model should make it possible to achieve the quality requirements of a service. The first two problem cases in this research, A and B, concentrated on achieving quality in service engineering. In the problem case A, the QRF method developed made it possible to capture the software and service requirements (including quality) into the architecture, thus concentrating on the RE and design phases in the service engineering. The method enabled the achievement of the common and variable requirements of several products, but the context was closed, concentrating on a product line inside a single company. The RE method developed in the problem case B transformed the RE from the product line context to the context of the digital service ecosystem. The new RE method for ecosystem based service engineering specified the RE phases, and described how the ecosystem assets are utilised in each RE phase. Furthermore, the required ecosystem elements for engineering services were identified, e.g. the actors with their responsibilities, knowledge management models, supporting services and existing digital services were identified. Thus, the common assets of the ecosystem substantially influence service engi-

neering. Therefore different kinds of business models and service engineering models are required when acting in an ecosystem.

The second objective of this research was to examine how to transform the quality evaluation from software and services to data quality evaluation, and to understand the key phases for quality evaluation of open data. The problem cases C and D concentrated on open data and the quality of open data. In the problem case C, the open data was inspected first from the business viewpoint to understand the nature, characteristics and the value of the open data that it provides for different actors. The concept of open data based business ecosystem was specified, which describes how to make business with data, including actors and their roles in the ecosystem, required support for business model elements from the ecosystem, and required support services for acting in data based business. In the problem case D, it was understood that the quality evaluation of data differs significantly from the quality evaluation of software. Although some of the same quality attributes were used in both cases, they all have different definitions, and means to achieve them. Furthermore, it is the responsibility of the data user to evaluate this. In the problem case D, a solution was provided on how to evaluate and utilise the quality of open data in a company's processes (e.g. the design or product/service development process), including the different evaluation phases and viewpoints on data quality.

The third and main objective of this research was to provide a model that combines ecosystem-based service engineering and open data quality evaluation under the same ecosystem concept. In problem case E, a new concept, the EODE, was created, that combines the differences and commonalities of digital service ecosystems, and open data ecosystems, under the same concept, enabling the actors of both ecosystems to act under the same ecosystem regulations. Thus, the EODE brings together and adapts the results of the problem cases A, B, C and D, and extends the knowledge management models and support services further to support data quality certification (see Figure 11).

The EODE fulfils the identified features for open data based digital service ecosystem described in Table 4. The developed service engineering model enables service co-innovation and co-development together with other ecosystem members utilising common ecosystem assets. The knowledge of the ecosystem (in the form of the domain models, quality policies, ontologies and design time artefacts) is utilised in digital service engineering. For all the ecosystem actors, the EODE provides an enabling environment with knowledge management models and support services for acting in the ecosystem, and providing digital services and quality certified open data in the ecosystem. The EODE supports each actor in defining their own business model, providing support for open business model elements, such as for finding partners, data, services and customers, and for contract making, and for marketing services and data. The EODE supports the quality evaluation of open data, both for the ecosystem, and for independent data consumers, such as digital service providers. Finally, the governance and regulation actions of the ecosystem support the bi-directional communication between digital service providers and open data providers.

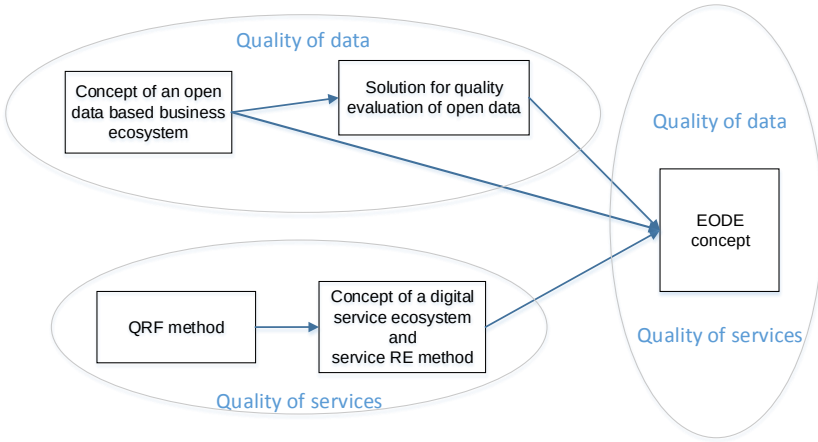

Figure 11. The combined results of this research.

The different application domains of the problem cases, such as digital healthcare, multimedia, and transportation, enabled examination of the service and data quality in specific usage contexts. Thus, the qualification of the research was performed with the help of these specific cases. The generalisation of the results from a cross domain perspective to general level was performed in each case, and finally generalisation from a cross domain perspective to the general level created the EODE concept. EODE is thus generic and applicable to any domain. The EODE can be applied to specific domains with the help of integration framework (i.e. the core) that provides domain models and domain knowledge models as services to the ecosystem. In comparison to Figure 1, the problem cases A and B enabled the solution to ensure quality in service engineering. Problem case C made it possible to reach the open data ecosystem from the business viewpoint, and problem case D concentrated on data quality evaluation. Thus, the EODE combines the viewpoints of business, big data, and data quality with service engineering in the ecosystem context.

5.2 Theoretical contribution

This dissertation provides new knowledge about quality-driven service engineering (Publication I), digital service engineering (Publication II), digital service ecosystems (Publication II), open data based business ecosystems (Publication III) and open data quality evaluation (Publication IV), and introduces new concepts (Publication V) that combines and extends all these subjects.

Capturing quality in ecosystem based digital service engineering

As there are multiple stakeholders involved in software or service engineering with different goals, traditional quality measures are often inadequate both for engineering service quality requirements, and for making architectural decisions. Different stakeholders may have different requirements for values of the quality indicators, and also different definitions and comprehension of the quality attributes and metrics. Therefore, quality attributes must be approached from a more global perspective, starting from the stakeholders' requirements elicitation, analysis and specification, and aiming at the architectural representation through compromising the conflict requirements and trade-off analysis. The comprehensive literature survey conducted revealed that the recently developed quality analysis methods have several shortcomings that do not make it possible to capture quality requirements and transfer them into the software or service architecture. The QFR method represented in this work filled the gap between requirements engineering and architectural modelling, and enabled tracing of design decisions to requirements, and vice versa. Furthermore, the QRF method was applicable to product lines, enabling separation of commonalities and variabilities among product line members. The application of the method to different case studies showed that the method worked according to its purpose.

In the case of digital service ecosystems, the ecosystem elements, i.e. the actors, the capabilities of the ecosystem, the knowledge management model and the existing services (both digital services and support services) influence the service engineering, especially the service requirements engineering. The digital services are innovated and developed together with ecosystem members, when the value is formed in a value network. When compared to traditional service engineering, new models and methods are required in the ecosystem, that enable this cooperation and the utilisation of the ecosystem assets, at the same time regulating and managing the service engineering in a way that the new digital service is ensured to be acceptable to the ecosystem. According to the literature review, there were no requirements engineering methods or service engineering models applicable to the digital service ecosystem. Furthermore, the concept of digital service ecosystem and its elements that assist in service engineering were not properly defined in the literature. Some methods and approaches could fill some part of the ecosystem-based service engineering, but they often concentrated only on their own viewpoints, and did not work together. The concept of digital service ecosystem was specified with the help of the existing definitions of digital ecosystem and service ecosystem, and the identified requirements of the ecosystem-based service RE method. The RE method represented in this work was the first RE method suitable for digital service ecosystems, which enables and guides the service co-innovation and co-development in an ecosystem environment. This work also specified the roles and responsibilities of the required ecosystem elements for service engineering. The method validation in large European projects revealed its benefits and potential applicability both as an RE method and as a means for communication among members of the ecosystem.

The RE method transferred the first parts of the QRF method from the product line context to the new, digital service ecosystem context. The last phase of the QRF method; quality representation, is performed in the same way in the ecosystem context with the help of knowledge management models. Thus, together these two methods provide a means to ensure that digital services capture their quality requirements in the ecosystem. The ecosystem capability model, and the knowledge management models and support services that enable the implementation of activities of the capability model, were specified to support the ecosystem based digital service engineering.

Evaluating and managing the quality of open data

The term 'open data based business ecosystem' was relatively new at the time when the research was conducted. Business ecosystems already had general, commonly accepted definitions and a relatively long history. Open data was not considered to be the target or the main resource of an ecosystem, but it was introduced in the form of data portals (e.g. http://ckan.org/), or innovation platforms (e.g. Riedl *et al.*, 2009). Therefore, it was natural to approach open data from the business viewpoint. Data value chains with known actors and data based business models had already been suggested in the literature, and open source communities existed. However, in this research, the concept of an open data based business ecosystem was represented that combined the actors of open data to the business environment. The first support services to enable data based businesses were specified (e.g. tool support for providing data and for developing services/applications and the services for finding, validating and adapting the data), and the required, new actors and actor roles were identified (e.g. the new actor; the Data Broker, and the new roles for support service providers, data providers and infrastructure and tool providers). The concept was generic, and thus applicable to any application domain. The concept creation with the help of industry representatives enabled it to capture the actual needs of the actors in the different domains. The validation of the concept with the same industry representatives enabled validation of the analysis results of the industry interviews, as well as estimation of the validity and the feasibility of the created concept.

The quality of data was identified as one of the obstacles for open data utilisation in the interviews with the industrial representatives. Many approaches for data quality evaluation exist that tend to evaluate common quality attributes, such as relevancy, believability, accuracy and popularity. Although a lot of work has been done for data quality attributes and metrics, there is no agreement on how to apply them to different contexts and situations. The solution for open data quality evaluation contributed in this research controls the data quality evaluation and management with the help of data quality policies. Although some promising policy-based approaches already exist for quality evaluation (Bizer and Cyganiak, 2009; Bertino and Lim, 2010; Rahman, Creese and Goldsmith, 2011), their practical application has not been demonstrated. Furthermore, they were not adaptable to the company's business processes, i.e. they did not address the different phases,

and questions relating to when and what evaluated data is required. The introduced solution in this research was suitable to the business processes of a company, and adaptable to the different needs and decision making points of the company. This was achieved with data quality policies that can be applied to different situations and contexts. The policies handle variability in the quality of data in the following ways: a) The target of quality attributes – certain quality attributes are applicable only to certain types of data sources; b) the applicability of attributes – the quality attributes are applicable in certain evaluation phases or data processing phases (i.e. data extraction processing, analysis and decision making); c) the target of quality metric – the different metrics are used to evaluate quality attributes, depending on the type of data source, and d) the applicability of the quality metric – different metrics are used to evaluate the attribute in different evaluation phases.

Quality of digital services and open data in an ecosystem

The developed concept of an open data based business ecosystem provided an environment to operate in the open data based business, whereas the solution for open data quality evaluation provided the settings and required elements for the open data quality evaluation. Combining these to the digital service ecosystem context, the elements that support capturing the quality requirements in digital service engineering (as the results of the QRF and service RE methods) are also extended to support open data based businesses, and data quality evaluation. The actors of the digital service ecosystem and open data based business ecosystem were merged, and the new resource, open data, was added to the ecosystem elements as an open data service that can be utilised in digital services. The ecosystem capability model with the related actions was extended to support the quality evaluation of the open data, and the business of the new actors. Furthermore, knowledge management models were extended to include quality policies to evaluate and to manage the quality of data. These policies were specified from the ecosystem viewpoint, and also from the viewpoints of the two main actors; the digital service provider and the open data providers. Furthermore, the initial taxonomy of the support service to implement the actions of the capability model were specified. This new digital service ecosystem concept, the EODE, is the first kind of ecosystem that makes it possible to engineer digital services that capture their quality requirements and utilise quality certified open data.

Summary of contributions

The contributions of this dissertation are summarised in Table 6 according to the intersection of the main entities of this research described in Figure 2 of subchapter 1.3.2.

Table 6. Summary of contributions

Research target	Contribution	Description
Digital service ecosystem	QRF method	A method for specifying quality requirements, transforming them into design decisions and representing them in architectural models
	RE method of digital services	A method for co-innovation and co-development of digital services in ecosystem utilising common assets and knowledge
	Concept of digital service ecosystem	Specification of the elements (i.e. members, capabilities, infrastructure with support services and knowledge management models, and digital services) of the ecosystem that enable the cooperation of ecosystem members, and co-innovation and co-development of digital services.
Data-based digital services	A solution for quality evaluation of open data	Specification and implementation of a solution for evaluating and managing the quality of open data with the help of data quality policies
Open data based business ecosystem	Concept of open data based business ecosystem	Specification of the elements (i.e. members, support services and required support for business model elements) of the ecosystem that enable the actors of open data to cooperate
	Quality certification process of open data	Specification of a process for certifying the quality of open data for an ecosystem
Data-based digital service engineering in ecosystems	Concept of an open data based digital service ecosystem	Specification of the members' roles, capability model, support services, knowledge management models, and core elements of the ecosystem that enable the actors of open data to cooperate in digital service ecosystems

5.3 Implications for new practices

Open data providers can be divided into three groups, according to their motives:

- Organisations that provide data for free, without any conditions or with some licenses that restrict the use of data. These usually include public administrations or other public entities that have a lot of data, but no abilities or resources to use the data in the form of data refinement or service development with the data. The data itself is usually strict and highly regulated.

- Individuals who provide data freely available to the Internet. These people are not interested in utilising the data, but the data provides business benefits for private companies. This data is commonly very heterogeneous, variable and unstructured.
- Organisations that do business from selling access to the data. These usually include private companies that provide access to data for paying customers. This data can be their own data, or refined or processed data of the other data providers

Open data as such is not valuable. Especially data from the social media is usually tangled and unreliable, having no meaning for further usage. Data refinement processes and analyses the data, thereby increasing the understanding and value of the data. Data certification adds the conformance on the data, meaning that the data is assessed, and it is seen as valuable for some purposes. Thus, data refinement and certification changes the data to an immaterial artefact, which enable people to do business with data. The original idea of open data, "data should be freely available for everyone to use and republish as they wish", is not necessarily valid in the case of open data based business; in some cases the data usage may be managed with usage fees or with data licences. However, data-based businesses have been identified as having sufficient potential to enable different kinds of ecosystems to emerge, including open data based digital service ecosystems.

Transforming to the ecosystem based business

The EODE concept described in this research enables the different actors of data and digital services to act in an open co-development environment, utilising common ecosystem assets, adhering to common ecosystem regulation assets, but still making their own business decisions. In addition to describing the EODE concept, this dissertation also revealed the benefits and significance of such an ecosystem for the different actors. The movement from the proprietary software or service business to openness is the key issue when considering joining in a digital service ecosystem such as EODE. The business model elements are affected by the ecosystem, and the transformation to a new business model should therefore be carefully evaluated. The benefits, potential and risks of starting an ecosystem-based business should be carefully estimated, and each actor should find its own role in the ecosystem. This research revealed the benefits and motives for different actors to join in the ecosystem, but also indicates the risks that should be considered.

Generally, there are also two options for the content of the digital service ecosystem: 1) The ecosystem emerges around a certain domain, being domain-dependent, or 2) there is a universal ecosystem applicable to all interested parties from different domains, being therefore generic, i.e. domain-independent. Furthermore, open data is often domain-specific, i.e. traffic data or environmental data, but applied in digital services of different domains. Thus, the same, domain-specific data can be included as a part of ecosystems of different application do-

mains. However, the domain models of the ecosystem may regulate the data and the usage of data in different ways in different domains. It is obvious that there is a great amount of work needed to establish and maintain the EODE ecosystem. The foundation of such an ecosystem requires a lot of investment first, but the more actors join, the more beneficial and relevant the ecosystem will be. The ecosystem requires a stakeholder that is responsible for the management, support, marketing and maintenance of the ecosystem, i.e. the ecosystem provider. The ecosystem provider can be, for example, a separate actor, or one of the ecosystem actors can take this role. The strictly defined responsibilities and rights of the different ecosystem actors guarantee that conflicts between the ecosystem members can be avoided, and that cooperation proceeds smoothly. The EODE has governance and regulation actions for directing, monitoring and managing the ecosystem that involves all the ecosystem members. These include, for example, the rules for establishment of trusted collaboration, the interaction rules, and the rules for joining and leaving the ecosystem. The ecosystem should validate each member when joining the ecosystem; thus, the trustworthiness of the members is already confirmed when cooperations are established, and contracts are made with different members of the ecosystem.

Acting in an ecosystem

The EODE enables different actors to perform different actions, supporting the businesses of the actors. For digital service providers, the ecosystem provides the models, methods, templates and guidelines that assist in service engineering. After the service provider has joined the ecosystem and made the transformation to the new kind of open service engineering model, all the utilities of the ecosystem are available. The service provider can utilise the existing assets and also the open data in his/her digital service, the quality of which is certified by the ecosystem. The ecosystem assists in matching required data quality with provided data quality of open data. The ecosystem also assists in finding appropriate data, partners, and in marketing the digital services.

For open data providers, the EODE provides the guidelines, supporting services and knowledge management models to assist in providing data. Data providers achieve several benefits when providing data to an ecosystem, such as more users for the data, and the more efficient marketing and promoting of the data. The ecosystem also encourages the data providers to ensure and improve the quality of the data; the ecosystem's data filtering policy makes it possible to bring to the ecosystem only data of which the quality reaches the required minimum values of the quality attributes of the data filtering policy. The filtering policy also monitors the quality of data during usage.

The support services of the EODE include eight main categories; utility services, data matchmaking services, monitoring and evaluation services, recognition services, adaptation services, open data analysis, visualisation services, and tool services. These categories of services provide several possibilities and business potential in EODE for other support service providers as well. The support service

providers are independent service providers that provide services for the usage of the ecosystem and the digital service providers. The service related pricing models are applicable to open data services, digital services and support services in the ecosystem. For example, in the pay-per-use pricing model (Weinhardt *et al.*, 2009), the customer pays only for the service usage. In the subscription model (Weinhardt *et al.*, 2009), the client pays a fixed price for a certain time frame when the service can be used. Internet related pricing models, such as Flickr multiple revenue stream model (Teece, 2010), freemium model (Teece, 2010) and free trial model (Gaudeul, 2010), are also applicable to digital services.

Ensuring quality in an ecosystem

The key concept of the EODE is to assist in achieving the quality of services and open data in the ecosystem based service engineering. For digital service providers, the quality of service is ensured with the knowledge management models, including RE method, service engineering method and templates to capture the quality requirements, and supporting services. These assets assist and guide the service engineering for service providers to reach the quality. To utilise data in digital services, the service provider must first define its strategy for data utilisation, for example, specifying what data is to be utilised, and where, when and how the data is to be utilised. The strategy assists in defining the data quality policies, which describe how the data is utilised.

Quality evaluation and management is controlled by the quality policies in the ecosystem (Figure 12). Quality policies must be defined first, after which suitable data can be searched. The service providers can express their requirements for the data to the ecosystem. The ecosystem searches for the relevant data, and ensures the trustworthiness of the data with the help of data filtering policies, before it is accepted to the usage of the ecosystem (Figure 12). Evaluation policies are used to evaluate quality attributes, and the achieved value is compared with the required values in the filtering policy. The service provider can then ensure the applicability of the data by evaluating the quality of the data specific to the situation at hand. The ecosystem provides evaluation policies for the usage of the service provider, for example, in the form of policy templates, which the service provider can configure according to his or her own purposes. The quality policies of the ecosystem are based on the ecosystem's quality knowledge; the service provider does not have to know the quality evaluation metrics or techniques for data quality evaluation, but, depending on the type of data source, the applicable metrics and techniques for each data quality attribute are selected automatically. The evaluation can also be automated, for example the service matchmaking algorithm may perform the quality evaluation with the help of the quality policy defined by the service provider. After data evaluation, the service provider reaches the certified data applicable for the situation. This data is brought to the decision making according to the decision-making policies of the service provider. This policy assists in selecting the most relevant data, and weighting quality attributes at certain decision-making points. Thus, the quality policies of the EODE provide

the assets to the service provider to ensure the quality and the value of the open data utilised in digital services.

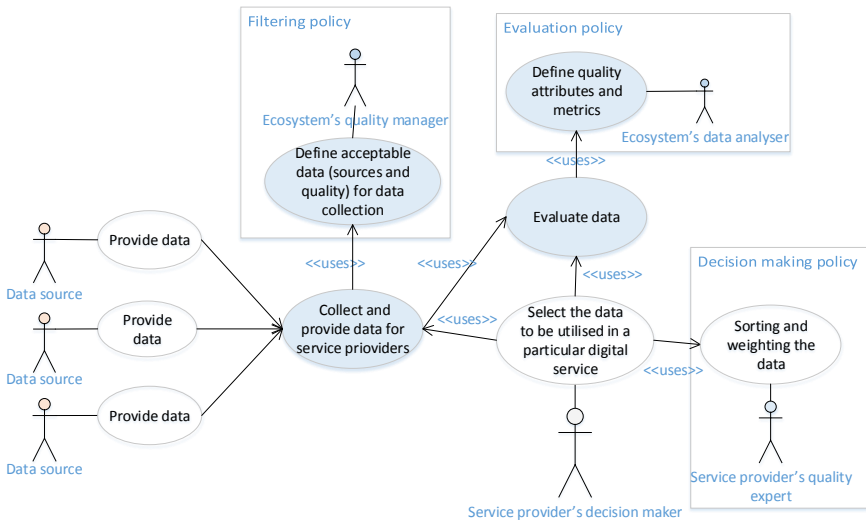

Figure 12. Management of open data quality in the ecosystem with the quality policies.

5.4 Scientific validity

This dissertation consists of five original publications that were published in peer reviewed scientific journals. Thus, the results presented in these publications have been reviewed and evaluated in high-quality scientific forums. Furthermore, the validity of the results is separately discussed in the original publications. Together these five publications form a consistent body of knowledge by specifying a new kind of collaboration environment for different actors of open data and digital services. The research results presented in the original publications are based on data achieved with the systematic mapping and empirical data. Systematic mapping tends to structure a research area by searching the literature in order to know what topics have been covered in the literature, and where the literature has been published (Petersen *et al.*, 2008). The empirical data is more human-related and achieved in several ways, such as with interviews, workshops and analytic methods.

According to Brewer (2000), validity of the conclusions drawn from the results must be evaluated in the light of the purposes for which the research was conducted. According to (Pekuri, 2013), testing the functionality of the developed construction is the means of providing the information needed in making the final conclusions. The purpose of the testing is to determine whether the construction works, therefore justifying either the formation of a technical norm or some other conclusions. Accordingly, testing also determines the validity of constructive re-

search (Pekuri, 2013). Easterbrook *et al* (2008) summarised four types of validity of empirical study: internal, external, construct and reliability. Internal validity measures whether the research was carried out correctly, whereas external validity is the degree to which it is warranted to generalise the results to other contexts. Construct validity evaluates whether the inferences made in the research are made correctly, i.e. whether the theoretical constructs are interpreted and measured correctly. Reliability focuses on whether the study yields the same results if other researchers replicate it.

Publication I presented research that resulted in a method for capturing quality requirements and Publication II in a method for requirements engineering of digital services. In both cases, the research started with theoretical analysis which was then continued with empirical analysis. The theoretical analysis resulted in the developed constructs, which were then validated and refined with the help of empirical analysis. The empirical data was collected with the help of interviews with relevant stakeholders from multiple sources, meetings and a questionnaire study. The trial usage of the methods helped to confirm that the methods worked as expected. The number of stakeholders trial-using the developed methods was high, which helped to support the internal validity of the research. In addition, the results were analysed and validated by several people; the author and the co-authors of the original publications. The trial-usage of both methods occurred among several cases in different domains, which supported the generalisability of the results and thus the external validity.

Publication III presented the research for the specification of the context of an open data based business ecosystem. The empirical data was gathered through industrial settings with the help of industry interviews and the data with the analysis results were reviewed and validated by the author and the co-authors of the original publication. The ecosystem concept was developed as a conclusion from the results of the literature analysis and the analysis of empirical data. The concept was then analysed and validated by several people: the author, co-authors of the original publications and the same company representatives that participated in the interviews, which supported the internal validity. The external validity was supported by the fact that the interviewees represented several domains, such as weather observation, media, healthcare and transport, which support the justification of the claim for the generalisability of the results.

Publication IV presented the solution for the quality evaluation of open data. In this case too, the empirical data was gathered with the help of industry interviews in two different domains and further the data and analysis results were reviewed and validated by several stakeholders. The literature analysis and the analysis of empirical data resulted in the specification of the solution. The implementation of the solution was conducted in collaboration with an industrial partner. An existing system of the partner was modified to enable its integration to the solution of data quality evaluation. Responsibilities in the development work were clearly divided to enable both organisations to focus on the development of their own software assets. The resulting solution and the results of its trial usage by the involved industrial partner were reviewed by several stakeholders from both organisations. The

co-development and co-validation of the solution supported the internal validity. The configurable quality policies of the solution supported its applicability to other cases and contexts, thus promoting external validity.

In all cases, construct validity was evaluated with the help of trial usage, theoretical testing in industrial settings or in multiple case studies. The trial usage resulted in identifying the functionality and applicability of the construct, and also the improvement requirements of the developed construct. The multiple sources of evidence in the form of several cases and several stakeholders supported the validity of the constructs.

Reliability of the research is supported by the proper documentation of the research, literature surveys, analysis results, interviews, questionnaires, developed constructs and included materials, process descriptions and guidelines. Furthermore, the constructs and the empirical cases were reviewed and analysed by several researchers in addition to the author, which provided reliability. However, the selection of companies and interviewees affects the results of the research, which means that the represented results cannot be generalised to all kinds of companies. In addition, in one case the selection of software professionals rather than business professionals could have resulted in more technical outcomes of the interviews.

Publication V presented the EODE concept that combines all the earlier research. All the developed constructs were validated in their own cases, which means that the internal validity was supported by having several stakeholders in analysis and validation, such as the author, co-authors, co-workers and several industrial representatives. The external validity is partly supported by the fact that the different parts of EODE were developed in different domains and their generalisability was evaluated separately in each case. Furthermore, generalisability in EODE is taken into account by separating the generic and domain-specific knowledge management models. The next step in external validation is the application of EODE to the different application domains.

5.5 Comparison to related work

The QRF method was developed for defining, representing and transforming quality requirements to architectural models, since no existing methods for R&A analysis committed themselves to quality requirements or specified how to transfer the quality requirements to architecture (Immonen and Niemelä, 2008). Some parts of the QRF method applied or extended existing approaches. The i* framework (Chung, Gross and Yu, 1999) was used to identify quality requirements from the viewpoints of different stakeholders. By refining the i* framework, it could be graphically represented which qualities were regarded as the most important and who the ‘owners’ of these qualities were. Furthermore, the Strategic Dependency model of the i* framework could be used to describe the variability of requirements between product family members (Immonen, 2006). The representation of R&A qualities in architectural models was implemented using the R&A profiles, which

were defined utilising the UML extension mechanisms (Aagedal *et al.*, 2004) that can be used for modelling certain quality aspects in architectural models. The abstraction levels of QADA (Matinlassi, Niemelä and Dobrica, 2002) were applied in separation of required and provided profiles. The required profile corresponded to R&A requirements, whereas the provided profile corresponded to the implemented R&A and could be later used in R&A analysis. For quality representation, the method utilises a stylebase (Merilinna, 2005) from which the architect can obtain information about the quality properties of each style and pattern and select the most suitable ones. The QRF method responds to the features of service co-development knowledge based service engineering from Table 4, being the first method describing how to capture quality into architectural models.

According to the literature review, no methods for ecosystem-based digital service engineering existed. Furthermore, the term digital service ecosystem was not yet specified when the research was started. The definition for the digital service ecosystem in this dissertation was developed with the help of the definition of digital ecosystem (Chang and West, 2006) and the definition of service ecosystems (Liu and Nie, 2009; Riedl *et al.*, 2009; Ruokolainen, 2013). In the work of Ruokolainen (2013), the service ecosystem engineering, especially the viability and sustainability of ecosystems, has been investigated. The work defines the service ecosystem engineering as a systems engineering life cycle which comprises the phases of analysis, design, instrumentation and operation of the ecosystem. The main elements of the EODE ecosystem in this dissertation conform to the ecosystem element classification of Ruokolainen; capabilities, members, services and infrastructure. However, in this dissertation, these elements are inspected from the digital service engineering viewpoint, not from the ecosystem engineering viewpoint. Therefore, the more detailed content of the main elements is specified to concentrate on actions, support services and knowledge management models required to support the ecosystem based requirements engineering of digital services. Thus, the RE method and ecosystem concept description support the features of service co-innovation, service co-development, knowledge based service engineering and enabling infrastructure from Table 4.

The term open data based business ecosystem was relatively new and not properly defined when the first research activities among open data were carried out. The existing value chains of data (Chen *et al.*, 2011; Kuk and Davies, 2011; Poikola, Kola and Hintikka, 2011; Tammisto and Lindman, 2011) were used as a starting point for identifying the actors of the ecosystem. Kuk and Davies (2011) introduced the assembly of complementarities involved in the chain from raw data to data-based services, including the parties that structure the raw data, make the data linkable, analyse or visualise the data, share the data within the source code of software and ultimately allow the developers to innovate services on top of the source code. Poikola, Kola and Hintikka (2011) defined the roles in the open data value chain from the data publishing perspective and the end-user's perspective, describing several roles in both classes. Tammisto and Lindman (2011) defined the roles of linked-data developers and application developers in a Finnish context. In addition, Chen *et al.* (2011) identified roles related to data analytics; Data-

as-a-Service (DaaS) providers collect, generate, and aggregate the content (i.e., data), and Analytics-as-a-Service (AaaS) providers deliver analytics services to analytics consumers. However, the industry interviews revealed that the existing actors and value chains are not adequate, but new actors and actor roles could be required. In the ecosystem, the value is created in networks rather than in chains. Furthermore, the interviews helped to identify services that are required to support the businesses of the actors. Therefore, the developed concept of an open data based business ecosystem is a new kind of collaboration environment that specifies the actors, supporting services and the business model elements of supporting open data based business.

The quality policies used in the quality evaluation of open data are based on the idea of the works of Bizer and Cyganiak (2009) and Rahman, Creese and Goldsmith (2011), where the information or the information sources concerning social media data are filtered according to some filtering policy and accepted according to decision making policy or a decision making function. However, the developed solution for quality evaluation of open data specifies the principles for data quality management for different types of data, such as social media data, feedback data, production data and market analysis data. These principles specify when, what and for whom the data is evaluated, and the configurable quality policies for data filtering, evaluation and decision making, taking into account the different types of data sources, different evaluation phases, and the data quality variability. These policies are configured and used when going through the different phases of the big data pipeline. In addition, the solution describes the data quality metadata that must be connected with the actual data, and also the data and quality metadata management in big data architecture.

The developed EODE concept is the first kind of concept that provides a new cooperating environment for the actors of digital service ecosystems and open data based business ecosystems. The more detailed content of the main elements; capabilities, members, services and infrastructure, are specified from the quality viewpoints of open data, starting from the required actions of the capability model and resulting in the required services and knowledge management models of the infrastructure to implement these actions.

5.6 Limitations of the research and future work

The research introduced in this dissertation combines four separately developed results together, creating the EODE concept, that assists in achieving the quality of services and open data in ecosystem based service engineering. Thus, the development and validation of the EODE concept has been carried out incrementally in several international and national research projects. Although several validation experiments of the different elements of EODE have been carried out, the whole EODE concept still requires more experimental tests, and empirical evaluations. The application of EODE to the different application domains and business fields is naturally the next step in the concept validation. By applying the concept

to certain domains (e.g. traffic, energy and healthcare), the content of the ecosystem elements will be specified in more detail in that specific domain. The domain model specifies the domain/application specific knowledge, and when using the domain model together with the generic knowledge management models, the service engineering can be adapted to the case at hand. The generic EODE concept with the KMM and service engineering models assist in defining the domain specific models.

The aim is to continue the research work in a large international research project in the future. The large number of project partners enables the project participants to act as ecosystem members, which also enables the cooperation and co-development of the ecosystem elements. Several actor roles can be identified, and their requirements can be examined, and the partner contributions can possibly implement some parts of the EODE concepts. International co-operation in an ecosystem also reveals the different kinds of national or legal aspects and regulations in different countries. In an ideal situation, some of the support services can be implemented, and by developing a user interface for configuring quality policies, the quality evaluation of open data services registered in the service registry can be implemented with the quality policies. The Digital Services Hub, which was used as the core of the EODE concept, is free to use for research and innovation purposes, and therefore can be used in future research projects as well.

The most important steps for the EODE validation currently include:

- Quality policy implementation at the ecosystem level. The current implementation supports quality evaluation inside a single company; the quality policies must be implemented and applied by the ecosystem, and also be applicable and configurable to certain situations of the service providers.
- Implementation of the support services that enable quality monitoring and automated quality evaluation in an ecosystem according to quality policies.
- An adaptable user interface for policy configuration and policy templates for defining the required quality.
- Implementation of a matching service, that automatically makes available the open data services, of which quality matches with the required quality.

Several future development targets were also identified after this research. Commonly, open data certification consists of legal, practical, technical and social aspects[14]. Currently, the EODE supports only open data quality certification, which can be considered as one of the technical and also practical aspects of the open data certification. However, the Open Data Certificates of the ODI currently does not consider quality in the way that is described in this dissertation, but relies on the data providers' voluntary documentation about the quality of data. The next step will be to extend the EODE with legal and practical aspects of the data certification. This is future work according to the following description:

[14] https://certificates.theodi.org (Accessed: 1 November 2016)

- Legal – Privacy: Data privacy aspects are domain-specific, and are often regulated differently in each country. The domain knowledge model must include the knowledge of how to take data privacy issues into account separately in each domain (e.g. healthcare, traffic, financial). Thus, when applying the EODE in different domains, privacy issues must be solved case by case. Currently, in EODE the trust between services and the data privacy is implemented by the Digital Service Hub in the context of personal data. The data owner has the data sovereignty, specifying the terms and conditions to use the data (Boris *et al.*, 2016). Data privacy must be extended to the case of open data, when the sovereignty of the data must be handled by the ecosystem.
- Legal – Licenses: Originally, licenses grant baseline rights to distribute copyrighted work. Ecosystem's filtering policy must be extended to ensure that data licenses applicable to the data are also applicable to the ecosystem. The data filtering policy may contain restrictions for license conditions that may prevent the data selection for the ecosystem, even if the quality of the data was ensured to be good enough. Currently, the dataset descriptions are linked with the concepts (in Concept Schema) that provide information about the data provider, and the nature of the data, and links to a more specific concept in the data ontology. In the next step, the concept schema will be extended and linked with the license ontologies likewise.
- Practical – Interoperability: In the Digital Services Hub, the basic service discovery is enabled by the human readable service description, and additional information associated to the service description. For more intelligent service discovery and, in particular, intelligent service matching, a semantic service data description is required. Semantically enriched descriptions support multilingual searches, matching different data elements describing the same thing, and using the relations of data elements in searching. Semantics also support interoperability between different services. Currently, the Digital Services Hub supports the transformation of the non-semantic data model to the semantic data model. In the next phase, the semantic service discovery and matching must be implemented in the EODE.
- Practical – Service availability and quality. The service availability is implemented in the Digital Services Hub in two ways; either the registered service notifies the Digital Services Hub regularly when active, or the Digital Services Hub continuously queries its services for availability. In addition to the information about the service availability, the EODE core must keep track and monitor the service quality, so that the potential service consumer may detect the available services, and also the current quality of the available services.

6. Conclusions

Digital service ecosystems provide several benefits to service providers, enabling service co-innovation and co-creation among ecosystem members utilising and sharing common assets and knowledge. Digital service engineering in an ecosystem requires new innovation practices, service engineering models and a new kind of collaboration environment that brings the benefits of the ecosystem available to service providers. Furthermore, the utilisation of open data in digital services requires new practices and evaluation models in order to ensure the quality and value of data. The dynamic characteristics of digital services and the utilisation of freely available open data in services cause new challenges for engineering reliable and trustworthy services.

This research concentrated on the quality of digital services in the context of the ecosystem. The main purpose of this research was to answer to the question of how to design the quality of digital services in open data and the ecosystem based service engineering. According to the state-of-the-art analysis, no such ecosystem currently exists that provides the required assets that make it possible to achieve the quality of services in service engineering, and to ensure the quality of open data utilised in digital services. Several features for that kind of ecosystem were identified in this dissertation, including the service co-innovation and cooperation model, knowledge to be utilised in service engineering, enabling infrastructure, support for transformation to open business models, evaluation methods for the quality of open data, and support for cooperation of actors of open data and digital services. As a result, the EODE concept specified in this work introduced a new cooperating environment, where the different actors of data based business and digital service providers can work together. The EODE implements the required features of the ecosystem according to the following;

- Service co-innovation and co-development: The EODE provides an ecosystem based service engineering model with the supporting method and templates that enable innovation of digital services together with different ecosystem members (e.g. business stakeholders, collaborator and customers) and co-development of services in a value network utilising common ecosystem assets.
- Knowledge based service engineering: The ecosystem gathers and manages knowledge (e.g. in the form of the domain models, quality policies,

ontologies and design time artefacts) and enables their utilisation in digital service engineering, and in quality evaluation of open data.

- Enabling infrastructure: The EODE provides the environment that enables the required actions of the ecosystem members. The environment includes knowledge management models, and support services for providing digital services and for evaluating the quality of open data, and the actions required for acting and cooperating in the ecosystem.
- Open business model: The EODE provides support for open business model elements, such as for finding partners, making contracts and marketing services, enabling the actors to transform towards open business models, which is helpful when acting in an ecosystem.
- Quality evaluation of open data: The EODE provides the data quality policies and quality evaluation methods for certifying the quality of open data for the ecosystem, and supports the quality evaluation of open data case-specifically for the digital service providers.
- Support for cooperation of the actors of open data and digital services: The governance and regulation actions of the EODE support bi-directional communication between the actors of the ecosystems, including, among others, trust-making inside the ecosystem, and the clear definition of responsibilities of the different actors.

The development and validation of the EODE concept was performed incrementally in several international and national research projects. The EODE concept is generic and applicable to any domain; the main idea of the EODE is that generic and domain specific knowledge can be kept separated. The generic capability and knowledge management models can be smoothly exploited together with the domain-specific models in different application domains when engineering and running digital services.

This research also described the required validation of the concept, and identified future research work and the development targets. More validation is required, especially to test the whole EODE concept in different applications and business fields. For example, by implementing the selected support services, developing a user interface for configuring quality policies and for evaluating the quality of open data services registered in the ecosystem's service registry, the applicability of the EODE to certain situations can be validated. Furthermore, the purpose is to extend the EODE with other aspects of data certification, such as legal and practical aspects.

References

Aagedal, J. O., de Miguel, M. A., Fafournoux, E., Lund, M. S. and Stolen, K. (2004) *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms*, *Technical Report 2004-06-01. Object Management Group*.

Ackoff, R. L. (1989) 'From data to wisdom', *Journal of Applied Systems Analysis*, 16, pp. 3–9.

Agichtein, E., Castillo, C., Donato, D., Gionis, A. and Mishne, G. (2008) 'Finding high-quality content in social media', in *International Conference on Web Search and Data Mining WSDM '08*. Palo Alto, USA, pp. 183–194.

Al-Fataftah, I. A. and Issa, A. A. (2012) 'A Systematic Review for the Latest Development in Requirement Engineering', *World Academy of Science, Engineering and Technology*, 6, pp. 691–698.

Allee, V. (2008) 'Value network analysis and value conversion of tangible and intangible assets', *Journal of Intellectual Capital*, 9(1), pp. 5–24.

Antunes, F. and Costa, J. P. (2012) 'Integrating decision support and social networks', *Advances in Human-Computer Interaction*, 2012(Article 9).

Auer, S. R., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z. (2007) 'DBpedia: A Nucleus for a Web of Open Data', *Semantic Web. Lecture Notes in Computer Science*, 4825, pp. 722–735.

Baden-Fuller, C. and Morgan, M. S. (2010) 'Business Models as Models', *Long Range Planning*, 43, pp. 156–171.

Behkamal, B., Kahani, M., Bagheri, E. and Jeremic, Z. (2014) 'A Metrics-Driven Approach for Quality Assessment of Linked Open Data', *Journal of Theoretical and Applied Electronic Commerce Research*, 9(2), pp. 64–79.

Bengtsson, P. O. and Bosch, J. (1998) 'Scenario-Based Architecture Reengineering', in *The fifth International Conference on Software Reuse*. Victoria, Canada, pp. 308–317.

Bennett, K., Layzell, P. J., Budgen, D., Brereton, L., Macaulay and Munro, M. (2000) 'Service-Based Software: The Future of Flexible Software', in *Proceedings of the Asia-Pacific Software Engineering Conference*. Singapore: IEEEE Computer Society Press, pp. 214–221.

Bertino, E. and Lim, H.-S. (2010) 'Assuring Data Trustworthiness - Concepts and Research Challenges', in Jonker, W. and Petković, M. (eds) *Secure Data*

Management. SDM 2010. Lecture Notes in Computer Science 6358. Berlin, Heidelberg: Springer, pp. 1–12.

Bhatia, S., Li, J., Peng, W. and Sun, T. (2013) 'Monitoring and analyzing customer feedback through social media platforms for identifying and remedying customer problems', in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Niagara, Canada, pp. 1147–1154.

Bizer, C. (2007) *Quality-driven information filtering in the context of web-based information systems. Ph.D. Thesis.* Berlin: Freie Universitet.

Bizer, C. and Cyganiak, R. (2009) 'Quality-driven information filtering using the WIQA policy framework', *Web Semantics: Science, Services and Agents on the World Wide Web archive*, 7(1), pp. 1–10.

Blackstone, A. (2012) *Sociological Inquiry Principles: Qualitative and Quantitative Methods*, *Flat World Knowledge Online Textbook*. Available at: http://2012books.lardbucket.org/books/sociological-inquiry-principles-qualitative-and-quantitative-methods/index.html (Accessed: 8 December 2016).

Blau, B., Krämer, J., Conte, T. and van Dinther, C. (2009) 'Service Value Networks', in *IEEE Conference on Commerce and Enterprise Computing*. Vienna, Austria, pp. 194–201.

Bodnar, T., Tucker, C., Hopkinson, K. and Bilen, S. (2014) 'Increasing the veracity of event detection on social media networks through user trust modeling', in *IEEE International Conference on Big Data*. Washington, USA, pp. 636–643.

Bollen, J., Mao, H. and Zeng, X. (2011) 'Twitter mood predicts the stock market', *Journal of Computational Science*, 2(1), pp. 1–8. doi: 10.1016/j.jocs.2010.12.007.

Boris, O., Auer, S., Cirullies, J., Jürjens, J., Menz, N., Schon, J. and Wenzel, S. (2016) *Industrial Data Space: Digital Souvereignity Over Data*, *Technical report*. Fraunhofer-Gesellschaft. doi: 10.13140/RG.2.1.2673.0649.

Bosch, J. (2009) 'From Software Product Lines to Software Ecosystems', in *The 13th International Software Product Line Conference (SPLC'09)*. San Francisco, USA, pp. 111–119.

Brewer, M. (2000) 'Research Design and Issues of Validity', in Reis, H. and Judd, C. (ed.) *Handbook of Research Methods in Social and Personality Psychology*. Cambridge: Cambridge University Press, pp. 3–16.

Cai, L. and Zhu, Y. (2015) 'The Challenges of Data Quality and Data Quality Assessment in the Big Data Era', *Data Science Journal*, 14(2), pp. 1–10.

Castañeda, V., Ballejos, L., Caliusco, L. and Galli, R. (2010) 'The Use of

Ontologies in Requirements Engineering', *Global Journal of Researches in Engineering*, 10(6), pp. 2–8.

Castillo, C., Mendoza, M. and Poblete, B. (2011) 'Information credibility on twitter', in *The 20th International Conference on World Wide Web*. Hyderabad, India, pp. 675–684.

Chae, B. (2015) 'Insights from Hashtag #SupplyChain and Twitter Analytics: Considering Twitter and Twitter Data for Supply Chain Practice and Research', *International Journal of Production Economics*. Elsevier, 165, pp. 247–259. doi: 10.1016/j.ijpe.2014.12.037.

Chan, C. M. L. (2013) 'From open data to open data innovation strategies: Creating E-Services Using Open Government Data', in *The 46th Hawaii International Conference on System Sciences (HICSS)*. Wailea, USA, pp. 1890–1899.

Chang, E. and West, M. (2006) 'Digital EcoSystems a next generation of collaborative environment', in *The 8th International Conference on Information Integration and Web-Based Applications & Services*. Yogyakarta, Indonesia, pp. 3–23.

Chen, Y., Kreulen, J., Campbell, M. and Abrams, C. (2011) 'Analytics Ecosystem Transformation: A force for business model innovation', in *Annual SRII Global Conference*. San Jose, CA, pp. 11–20.

Chenyun, D., Lin, D., Kantarcioglu, M., Bertino, E., Celikel, E. and Thuraisingham, B. (2009) 'Query Processing Techniques for Compliance with Data Confidence Policies', in *The 6th VLDB Workshop on Secure Data Management*. Lyon, France, pp. 49–67.

Chesbrough, H. W. and Appleyard, M. M. (2007) 'Open innovation and strategy', *California Management Review*, 50, pp. 57–76.

Chung, L., Gross, D. and Yu, E. (1999) 'Architectural design to meet stakeholders requirements', in *The 1st Working IFIP Conference on Software Architecture*. San Antonio, USA, pp. 545–564.

Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. (2000) *Non-functional requirements in software engineering*. Boston, Dordrecht: Kluwer Academic Publishers.

Cortellessa, V. and Grassi, V. (2007) 'Reliability Modeling and Analysis of Service-Oriented Architectures', in *Test and Analysis of Web Services*. Springer-Verlag (LNCS), pp. 339–362.

Cortellessa, V. and Pompei, A. (2004) 'Towards a UML profile for QoS: a contribution in the reliability domain', in *The 4th international workshop on Software and performance*. Redwood Shores, USA, pp. 197–206.

Cortellessa, V., Singh, H. and Cukic, B. (2002) 'Early reliability assessment of UML based software models', in *The 3rd International Workshop on Software and Performance*. Rome, Italy, pp. 302–309.

Creswell, J. W. and Plano Clark, V. L. (2007) *Designing and conducting mixed methods research*. CA: Sage Publications: Thousand Oaks.

Crnkovic, G. D. (2010) 'Constructive Research and Info-computational Knowledge Generation', in Magnani, L., Carnielli, W., and Pizzi, C. (eds) *Model-Based Reasoning in Science and Technology*. Berlin, Heidelberg: Springer (Studies in Computational Intelligence), pp. 359–380. doi: 10.1007/978-3-642-15223-8.

Dai, C., Lin, D., Bertino, E. and Kantarcioglu, M. (2008) 'An approach to evaluate data trustworthiness based on data provenance', in Jonke, W. and Petkovic, M. (eds) *SDM 2008. Lecture Notes on Computer Science 5159*. Springer, pp. 82–98.

Davis, R. N. (1999) 'Web-based administration of a personality questionnaire: Comparison with traditional methods', *Behavior Research Methods, Instruments, & Computers*, 31(4), pp. 572–577. doi: 10.3758/BF03200737.

Dobrica, L. and Niemelä, E. (2000) 'Attribute-based product-line architecture development for embedded systems', in *The 3rd Australasian Workshop on Software and Systems Architectures*. Sydney, pp. 76–88.

Dobrica, L. and Niemelä, E. (2002) 'A Survey on Software Architecture Analysis Methods', *IEEE Transactions on Software Engineering*, 28(7), pp. 638–653.

Dobson, G. and Sawyer, P. (2006) 'Revisiting Ontology-Based Requirements Engineering in the age of the Semantic Web', in *International Seminar on Dependable Requirements Engineering of Computerised Systems*. Halden: Institute for Energy Technology (IFE).

Dorfman, M. and Thayer, R. H. (1997) *Software Requirements Engineering*. Los Alamitos, USA: IEEE Computer Society Press.

Easterbrook, S., Singer, J., Storey, M.-A. and Damian, D. (2008) 'Selecting Empirical Methods for Software Engineering Research', in *Guide to Advanced Empirical Software Engineering*. London: Springer London, pp. 285–311. doi: 10.1007/978-1-84800-044-5_11.

Erl, T. (2007) *SOA Principles of Service Design*. New Jersey, USA: Prentice Hall.

Fabijan, A., Holmström Olsson, H. and Bosch, J. (2015) 'Customer Feedback and

Data Collection Techniques in Software R&D: A Literature Review', *Lecture Notes in Business Information Processing*, 210, pp. 139–153.

Ferrando-LIopis, R., Lopez-Berzosa, D. and Mulligan, C. (2013) 'Advancing value creation and value capture in data-intensive contexts.', in *IEEE International Conference on Big Data*. Silicon Valley, USA, pp. 5–9.

Fricker, S. (2010) 'Requirements Value Chains: Stakeholder Management and Requirements Engineering in Software Ecosystems', in *Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science, Vol. 6182*, pp. 60–66.

Gaudeul, A. (2010) 'Software Marketing on the Internet: The Use of Samples and Repositories', *Economics of Innovation and New Technology*, 19(3), pp. 259–281.

Gil, Y. and Artz, D. (2007) 'Towards content trust of web resources', *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4), pp. 227–239.

Gorton, I. and Klein, J. (2015) 'Distribution, Data, Deployment: Software Architecture Convergence in Big Data Systems', *IEEE Software*, 32(3), pp. 78–85.

Grassi, V. (2004) 'Architecture-based Dependability Prediction for Service-Oriented Computing', in *Proceedings of the Twin Workshops on Architecting Dependable Systems*. Edinburgh, Scotland: Springer-Verlag, pp. 279–299.

Grünbacher, P., Egyed, A. and Medvidovic, N. (2003) 'Reconciling Software Requirements and architectures with Intermediate Models', *Software and Systems Modeling*, 3, pp. 235–253.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A. and Khan, S. U. (2015) 'The rise of "big data" on cloud computing: review and open research issues', *Information Systems*, 47, pp. 98–115.

Heimstädt, M., Saunderson, F. and Heath, T. (2014) 'From Toddler to Teen: Growth of an Open Data Ecosystem', *eJournal of eDemocracy & Open Government (JeDEM)*, 6(2), pp. 123–135.

Henttonen, K., Matinlassi, M., Niemelä, E. and Kanstrén, T. (2007) 'Integrability and extensibility evaluation from software architectural models - a case study', *Open Software Engineering Journal*, 1(1), pp. 1–20.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) 'Design Science in Information System Research', *MIS Quarterly*, 28, pp. 75–105.

Hirsjärvi, S. and Hurme, H. (2001) *Tutkimushaastattelu: teemahaastattelun teoria ja käytäntö (in Finnish)*. Helsinki: Yliopistopaino.

HM Government Cabinet Office (2012) *Open Data White Paper: Unleashing the Potential*. London, UK.

Husnain, M., Waseem, M. and Ghayyur, S. A. K. (2009) 'An Interrogative Review of Requirement Engineering Frameworks', *International Journal of Reviews in Computing*, 2, pp. 1–8.

Iansiti, M. and Levien, R. (2004) 'Creating Value in Your Business Ecosystem', *Harvard Business Review*, March 2004, pp. 68–78.

Immonen, A. (2006) 'A method for predicting reliability and availability at the architecture level', in Käkölä, T. and Duenas, J. C. (eds) *Software Product Lines: Research Issues in Engineering and Management*. Berlin, Heidelberg: Springer, pp. 373–422.

Immonen, A. and Evesti, A. (2008) 'Validation of the reliability analysis method and tool', in *The 12th International Software Product Line Conference, 5th Software Product Lines Testing Workshop*. Limerick, Ireland, pp. 163–168.

Immonen, A. and Niemelä, E. (2008) 'Survey of reliability and availability prediction methods from the viewpoint of software architecture', *Software and Systems Modeling*, 7(1), pp. 49–65.

Immonen, A. and Pakkala, D. (2014) 'A survey of methods and approaches for reliable dynamic service compositions', *Service Oriented Computing and Applications*, 8(2), pp. 129–158.

Immonen, A., Palviainen, M. and Ovaska, E. (2013) 'Towards open data based business: Survey on usage of open data in digital services', *International Journal of Research in Business and Technology*, 4(1), pp. 286–295. doi: 10.0001/ijrbt.v4i1.197.

ISO (2008a) *ISO/IEC 25012- Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model*. Geneva, Switzerland: International Organization for Standardization.

ISO (2008b) 'ISO 9001:2008 Quality management systems - Requirements'. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2001) *ISO/IEC 9126-1: Software Engineering-Software product quality-Part 1: Quality model*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2003a) *ISO/IEC TR 9126-2: Software Engineering-Software product quality-Part 2: External metrics*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2003b) *ISO/IEC TR 9126-3: Software engineering-Software product quality-Part 3: Internal metrics*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2005) *Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*. Geneva, Switzerland: International Organization for Standardization.

Jansen, B. J., Zhang, M., Sobel, K. and Chowdury, A. (2009) 'Twitter power: Tweets as electronic word of mouth', *Journal of the American Society for Information Science and Technology*. John Wiley & Sons, Inc., 60(11), pp. 2169–2188. doi: 10.1002/asi.v60:11.

Jansen, S. and Cusumano, M. (2012) 'Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance', in *The 4th International Workshop on Software Ecosystems*. Cambridge, USA, pp. 40–58.

Jazayeri, M., Ran, A. and van der Linden, F. (2000) *Software Architecture for Product Families*. Boston, USA: Addison-Wesley.

Järvinen, P. (2004) *On research methods*. Tampere, Finland: Opinpajan kirja.

Kaiya, H. and Saeki, M. (2005) 'Ontology Based Requirements Analysis: Lightweight Semantic Processing Approach', in *The 5th International Conference on Quality Software (QSIC'05)*. Melbourne, Australia, pp. 223–230.

Kazman, R., Klein, M. and Clement, P. (2000) *ATAM: Method for Architecture Evaluation*, *The 4th IEEE International Conference on Engineering of Complex Computer Systems*. Monterey, USA: Carnegie Mellon University.

Khriyenko, O. (2012) 'Collaborative Service Ecosystem - Step Towards the World of Ubiquitous Services', in *Proceedings of the IADIS International Conference Collaborative Technologies*. Lisbon, Portugal., pp. 19–21.

Kimita, K., Akasaka, F., Shimomura, Y., Öhrwall Rönnbäck, A. and Sakao, T. (2009) 'Requirement Analysis for User-Oriented Service Design', *Asian International Journal of Science and Technology Production & Manufacturing Engineering*, 2(3), pp. 11–23.

Kotonya, G. and Sommerville, I. (1998) *Requirements Engineering: processes and techniques*. Wiley Publishing.

Kuk, G. and Davies, T. (2011) 'The Roles of Agency and Artifacts in Assembling Open Data Complementarities', in *The 32rd International Conference on Information Systems*. Shanghai, China, p. 16.

Leangsuksun, C., Shen, L., Liu, T., Song, H. and Scott, S. (2003) 'Availability

Prediction and Modeling of High Availability OSCAR Cluster', in *IEEE International Conference on Cluster Computing*. Hong Kong, pp. 380–386.

Lee, A. S. and Baskerville, R. L. (2003) 'Generalizing Generalizability in Information Systems Research', *Information Systems Research*. INFORMS, 14(3), pp. 221–243. doi: 10.1287/isre.14.3.221.16560.

Lehtiranta, L., Junnonen, J.-M., Kärnä, S. and Pekuri, L. (2015) 'The Constructive Research Approach: Problem Solving for Complex Projects', in Pasian, B. (ed.) *Designs, Methods and Practices for Research of Project Management*. Burlington, USA: Gower, pp. 95–106. Available at: http://www.gpmfirst.com/books/designs-methods-and-practices-research-project-management/constructive-research-approach (Accessed: 21 November 2016).

Lehto, I., Hermes, J., Ahokangas, P. and Myllykoski, J. (2013) 'Collaboration in cloud businesses – value networks and ecosystems', *Communications of the Cloud Software*.

Li, S. and Fan, Y. (2011) 'Research on the Service-Oriented Business Ecosystem (SOBE)', in *The 3rd International Conference on Advanced Computer Control (ICACC)*. Harbin, China, pp. 502–505.

Liu, P. and Nie, G. (2009) 'Research on Service Ecosystems: State of the Art', in *International Conference on Management and Service Science, MASS '09*. Wuhan, China, pp. 1–4.

Livesey, C. (2007) *Sociological Research Skills: Focused (Semi-structured) Interviews*. Sociology Central. Available at: http://www.sociology.org.uk/methfi.pdf.

Loniewski, G., Insfran, E. and Abrahão, S. (2010) 'A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development', *Model Driven Engineering Languages and Systems, Lecture Notes in Computer Science*, 6395, pp. 213–227.

Ludwig, T., Reuter, C. and Pipek, V. (2015) 'Social Haystack', *ACM Transactions on Computer-Human Interaction*. ACM, 22(4), pp. 1–27. doi: 10.1145/2749461.

Lyu, M. R. (1996) *Handbook of software reliability engineering*. New York, USA: Mcgraw-Hill.

Lähteenmäki, J., Leppänen, J. and Kaijanranta, H. (2008) 'Document-based service architecture for communication between health and wellness service providers and customers', in *The 2nd International Conference on Pervasive Computing Technologies for Healthcare, Pervasive Health 2008*. Tampere, Finland, pp. 275–278.

Ma, J. and Chen, H. (2008) ‘A Reliability Evaluation Framework on Composite Web Service’, in *IEEE International Symposium on Service-Oriented System Engineering*, pp. 123–128.

Madnick, S. E., Wang, R. Y., Lee, Y. W. and Zhu, H. (2009) ‘Overview and framework for data and information quality research’, *Journal of Data and Information Quality*, 1(1), pp. 1–22.

March, S. T. and Smith, G. F. (1995) ‘Design and natural science research on information technology’, *Decision Support Systems*, 15, pp. 251–266.

Matinlassi, M. (2004) ‘Comparison of software product line architecture design methods: COPA, FAST, FORM, KobrA and QADA’, in *The 26th International Conference on Software Engineering (ICSE 2004)*. Edinburgh, UK, pp. 127–136.

Matinlassi, M., Niemelä, E. and Dobrica, L. (2002) *Quality-driven architecture design and quality analysis method, A revolutionary initiation approach to a product line architecture*. Espoo, Finland: VTT Electronics.

Mendes, P. N., Mühleisen, H. and Bizer, C. (2012) ‘Sieve: Linked Data Quality Assessment and Fusion’, in *The 1st International Workshop on Linked Web Data Management (LWDM 2011) at the 15th International Conference on Extending Database Technology, EDBT 2012*. New York, USA: ACM Press. doi: 10.1145/2320765.2320803.

Merilinna, J. (2005) *A tool for Quality-Driven Architecture Model Transformation*. Espoo, Finland: VTT Publications.

Naumann, F. and Rolker, C. (2000) ‘Assessment methods for information quality criteria’, in *The 5th International Conference on Information Quality*. Boston, USA, pp. 148–162.

Niemelä, E., Immonen, A., Kanstren, T., Matinlassi, M., Merilinna, J. and Niskanen, A. (2005) ‘Quality Evaluation by QADA’, in *A half-day tutorial in the 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005.* Pittsburg, USA.

Niemelä, E., Kalaoja, J. and Lago, P. (2005) ‘Towards an Architectural Knowledge Base for Wireless Service Engineering’, *IEEE Transactions on Software Engineering*, 31(5), pp. 361–379.

Niemelä, E., Matinlassi, M. and Lago, P. (2003) ‘Architecture-centric approach to wireless service engineering’, in *Annual Review of Communications*, pp. 875–889.

Nurse, J. R. C., Agrafiotis, I., Creese, S., Goldsmith, M. and Lamberts, K. (2013) ‘Building Confidence in Information−Trustworthiness Metrics for Decision Support’,

in *The 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom−13).* Melbourne, Australia, pp. 535–543.

Nurse, J. R. C., Rahman, S. S., Creese, S., Goldsmith, M. and Lamberts, K. (2011) 'Information Quality and Trustworthiness: A Topical State-of-the-Art Review', in *International Conference on Computer Applications and Network Security (ICCANS)*. Male, The Maldives, pp. 492–500.

Nuseibeh, B. and Easterbrook, S. (2000) 'Requirements engineering: a roadmap', in *Future of Software Engineering, ICSE '00*. NY, USA, pp. 37–46.

OASIS (2006) *Reference Model for Service Oriented Architecture 1.0*. Organization for the Advancement of Structured Information Standards. OASIS. Available at: https://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html (Accessed: 6 November 2016).

Olkkonen, T. (1994) *Johdatus teollisuustalouden tutkimustyöhön.* Otaniemi, Finland: Aalto-yliopiston teknillinen korkeakoulu.

OMG (2002) *Unified Modeling Language (UML), version 1.5*. Object Management Group.

Osterwalder, A., Parent, C. and Pigneur, Y. (2004) 'Setting up an ontology of business models', in *The 16th International Conference on Advanced Information Systems Engineering (CAiSE03) Workshops*. Riga, Latvia, pp. 319–324.

Ovaska, E., Evesti, A., Henttonen, K., Palviainen, M. and Aho, P. (2010) 'Knowledge based quality-driven architecture design and evaluation', *Information and Software Technologies*, 52(6), pp. 577–601.

Ovaska, E. and Kuusijärvi, J. (2014) 'Piecemeal development of Intelligent Smart Space Applications', *IEEE Access*, 2, pp. 199–214.

Ovaska, E., Salmon Cinotti, T. and Toninelli, A. (2012) 'The Design Principles and Practices of Interoperable Smart Spaces', in Xiaodong, L. and Yang, L. (eds) *Advanced Design Approaches to Emerging Software Systems: Principles, Methodologies and Tools*. IGI GLobal, pp. 18–47.

Oyegoke, A. (2011) 'The constructive research approach in project management research', *International Journal of Managing Projects in Business*, 4(4), pp. 573–595. doi: 10.1108/17538371111164029.

Pantsar-Syväniemi, S., Purhonen, A., Ovaska, E., Kuusijärvi, J. and Evesti, A. (2012) 'Situation-based and self-adaptive applications for the smart environment', *Journal of Ambient Intelligence and Smart Environments*, 4(6), pp. 491–516.

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2007) 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*. M. E. Sharpe, Inc., 24(3), pp. 45–77. doi: 10.2753/MIS0742-1222240302.

Pekuri, L. (2013) *Perspectives on constructive research approach – In search of the basis for validation*. University of Oulu, Finland.

Perr, J., Appleyard, M. M. and Sullivan, P. (2010) 'Open for Business: Emerging Business Models in Open Source Software', *International Journal of Technology Management*, 52(3), pp. 432–456.

Petersen, K., Feldt, R., Mujtaba, S. and Mattsson, M. (2008) 'Systematic mapping studies in software engineering', *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*. BCS Learning & Development Ltd., pp. 68–77. Available at: http://dl.acm.org/citation.cfm?id=2227123 (Accessed: 4 May 2017).

Pham, T.-T. and Defago, X. (2013) 'Reliability Prediction for Component-based Software Systems with Architectural-level Fault Tolerance Mechanism', in *8th International Conference on Availability, Reliability and Security*. Regensburg, Germany, pp. 11–20.

Poikola, A., Kola, P. and Hintikka, K. A. (2011) *Public Data - an introduction to opening information resources*. Helsinki, Finland: Ministry of Transport and Communications. Available at: http://www.scribd.com/doc/57392397/Public-Data.

Purao, S., Rossi, M. and Sein, M. K. (2010) 'On Integrating Action Research and Design Research', in Hevner, A. and Chatterjee, S. (eds) *Design Research in Information Systems, Theory and Practise*. Springer USA, pp. 179–194. doi: 10.1007/978-1-4419-5653-8_13.

Rafique, I., Lew, P., Qanber Abbasi, M. and Li, Z. (2012) 'Information Quality Evaluation Framework: Extending ISO 25012 Data Quality Model', *World Academy of Science, Engineering and Technology*, 65, pp. 523–528.

Rahman, S. S., Creese, S. and Goldsmith, M. (2011) 'Accepting information with a pinch of salt: handling untrusted information sources', in *Security and Trust Management, Lecture Notes in Computer Science volume 7170*, pp. 223–238.

Ramesh, B. (2015) 'Big data architecture', *Studies in Big Data*, 11, pp. 29–59.

Reussner, R. H., Schmidt, H. W. and Poernomo, I. H. (2003) 'Reliability prediction for component-based software architectures', *Journal of Systems and Software*, 66(3), pp. 241–252.

Riedl, C., Böhmann, T., Leimeister, J. M. and Krcmar, H. (2009) 'A Framework for Analysing Service Ecosystem Capabilities to Innovate', in *The 17th European Conference on Information Systems (ECIS'09)*. Verona, Italy, pp. 2097–2108.

Rodrigues, G. N., Roberts, G., Emmerich, W. and Skene, J. (2003) 'Reliability Support for the Model Driven Architecture', in *The 2nd IEEE-ACM-SIGSaFT ICSE Workshop on Software Architectures for Dependable Systems (WADS"03)*. Portland, USA, pp. 79–98.

Ruokolainen, T. (2013) *A Model-Driven Approach to Service Ecosystem Engineering (PhD Thesis)*. Helsinki, Finland: University of Helsinki, Department of Computer Science.

Ruokolainen, T. and Kutvonen, L. (2009) 'Managing Interoperability Knowledge in Open Service Ecosystems', in *The 13th Enterprise Distributed Object Computing Conference Workshops*. Auckland, New Zealand, pp. 203–211.

Schindlholzer, B., Uebernickel, F. and Brenner, W. (2011) 'A Method for the Management of Service Innovation Projects in Mature Organizations', *International Journal of Service Science, Management, Engineering, and Technology*, 2(4), pp. 25–41.

Smidts, C. and Li, M. (2000) 'Software Engineering Measures for Predicting Software Reliability in Safety Critical Digital Systems'. University of Maryland, Washington D.C. (Tertiary Software Engineering Measures for Predicting Software Reliability in Safety Critical Digital Systems).

Sommerville, I. (2009) *Software Engineering*. 9th edn. Addison-Wesley.

Stathel, S., Finzen, J., Riedl, C. and May, N. (2008) 'Service Innovation in Business Value Networks', in *The 18th International RESER Conference*. Stuttgart, Germany, pp. 288–302.

Tammisto, Y. and Lindman, J. (2011) 'Open Data Business Models', in *The 34th Information Systems Seminar in Scandinavia*. Turku, Finland, pp. 762–777.

Teece, D. J. (2010) 'Business Models, Business Strategy, and Innovation', *Long Range Planning*, 43(2–3), pp. 172–194.

Trochim, W. M. (2006) *The Research Methods Knowledge Base, 2nd Edition*, *On-line book*. Available at: https://www.socialresearchmethods.net/kb/.

Vandermerwe, S. and Rada, J. (1988) 'Servitization of business: Adding value by adding services', *European Management Journal*, 6(4), pp. 314–324. doi: 10.1016/0263-2373(88)90033-3.

Wang, R. and Strong, D. (1996) 'Beyond Accuracy: What Data Quality Means to

Data Consumers', *Journal of Management Information Systems*, 12(4), pp. 5–33.

Weinhardt, C., Anandasivam, A., Blau, B. and Stosser, J. (2009) 'Business Models in the Service World', *IT Professional*, 11(2), pp. 28–33.

Wiesner, S., Peruzzini, M., Doumeingts, G. and Thoben, K. D. (2012) 'Requirements Engineering for Servitization in Manufacturing Service Ecosystems (MSEE)', in *Conference on Industrial Product Service Systems (CIRP IPS2 2012)*. Tokyo, Japan, pp. 291–296.

Xiang, J., Liu, L., Qiao, W. and Yang, J. (2007) 'SREM: A Service Requirements Elicitation Mechanism based on Ontology', in *The 31st Annual International Computer Software and Applications Conference, COMPSAC 2007*. Beijing, China, pp. 196–203.

Zhang, J. and Fan, Y. (2010) 'Current state and research trends on business ecosystem', in *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*. Perth, Australia, pp. 1–5.

Zhou, J., Ovaska, E., Evesti, A. and Immonen, A. (2011) 'OntoArch Reliability-aware Software Architecture Design and Experience', in Dogru, A. and Bicer, V. (eds) *Modern Software Engineering Concepts and Practices: Advanced Approaches*. New York, USA: IGI Global, pp. 48–74.

Publication I

Capturing quality requirements of product family architecture

Information and Software Technology,
Vol. 49, Issue 11–12, pp. 1107–1120.

Available online at www.sciencedirect.com
ScienceDirect
Information and Software Technology 49 (2007) 1107–1120
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

Capturing quality requirements of product family architecture

Eila Niemelä *, Anne Immonen

VTT Technical Research Centre of Finland, Software Architectures and Platforms, P.O.Box 1100, FIN-90571 Oulu, Finland

Received 4 May 2006; received in revised form 27 October 2006; accepted 5 November 2006
Available online 13 December 2006

Abstract

Software quality is one of the major issues with software intensive systems. Moreover, quality is a critical success factor in software product families exploiting shared architecture and common components in a set of products. Our contribution is the QRF (Quality Requirements of a software Family) method, which explicitly focuses on how quality requirements have to be defined, represented and transformed to architectural models. The method has been applied to two experiments; one in a laboratory environment and the other in industry. The use of the QRF method is exemplified by the Distribution Service Platform (DiSeP), the laboratory experiment. The lessons learned are also based on our experiences of applying the method in industrial settings.

Keywords: Quality requirement; Software architecture; Traceability; Software product family

1. Introduction

Decreased development cost and increased productivity are among the main benefits of product family engineering (PFE), which is currently often applied to software intensive systems. The goal in software PFE is to use as much as possible the same software assets, i.e. requirements, architecture, and components, in all family members. The goal is achieved with a systematic approach of maximizing software reuse and managing variability, which also assists in achieving software of high quality and maintaining a desired quality level of products.

Specifying requirements for a product family (PF) is a challenging task, and specifying quality requirements for a product family is even more challenging due to the varying quality properties required in different family members. Because of the long period of time that the family architecture is used as a basis of the family members, and since quality is the key issue in the family with its varying family members, it is a necessity

- to define quality requirements properly,
- to manage variability of quality properties, and
- to transform quality requirements to architectural models.

Several attempts have been made for defining and transforming quality requirements to architectures. Chung et al. [1,2] have introduced 'softgoals' as a means of capturing stakeholders' requirements and making tradeoffs between conflicting quality requirements. Softgoals are part of the NFR (Non-Functional Requirements) framework, which categorizes NFRs and provides a qualitative approach for dealing with them. The CBSP (Component Bus System Property) method applies intermediate models in order to reconcile functional requirements and architecture [3]. Neither of the approaches, however, provides a systematic approach to tracing quality requirements, including varying quality requirements, to the models of a product family architecture.

In model driven architecture development, the assumptions are that (1) if the family architecture is properly defined, the architecture of each family member can be derived from it using desired qualities, and (2), based on model transformation techniques, the source code of a

* Corresponding author. Tel.: +358 20 722 2228; fax: +358 20 722 2320.
E-mail addresses: Eila.Niemela@vtt.fi (E. Niemelä), Anne.Immonen@vtt.fi (A. Immonen).

0950-5849/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2006.11.003

family member can be generated from the specified architecture [4]. The QRF method introduced in this paper is a systematic method for eliciting and defining quality requirements, tracing and mapping these requirements to architectural models and for enabling quality evaluation at the early phases of PF development. Furthermore, the method supports defining variable quality properties for architecture modeling.

The QRF method was developed incrementally. First, it was applied in a laboratory experiment of defining evolution and execution qualities for the DiSeP family. Second, the method was applied to an industrial product family. Last, the method was refined and reapplied to the DiSeP family, which is also used as an example in this paper.

The structure of this paper is as follows. Section 2 surveys issues related to quality requirements traceability and family architecture development methods. Section 3 introduces an overview of the QRF method. Section 4 presents how the method was applied in a case example. Section 5 summarizes our lessons learned and a conclusion closes the paper.

2. Background

2.1. Related research

2.1.1. Quality attributes

Software quality is a degree of excellence regarding the ability of software to provide a desired combination of quality characteristics [5]. The software quality model [6] defines six categories of quality characteristics: functionality, reliability, usability, efficiency, maintainability, and portability. Quality characteristics are externally or internally observable properties of software systems, also called quality attributes [7].

Quality attributes can be further classified into two categories; functional qualities, which are observable at execution time (i.e. execution qualities), and non-functional qualities, which are observable during the product life cycle (i.e. evolution qualities) [8]. Functional qualities express themselves in the behavior of the system, while non-functional qualities are embodied in the static structures of software systems.

The interest of the quality attributes for software architecture is in how quality attributes interact with, and constrain, each other, and how they affect the achievement of other quality attributes (i.e. tradeoffs).

2.1.2. Capturing quality to architecture

The i∗ framework [1] helps to detect where the quality requirements originate and what kind of negotiations should take place. The NFR (non-functional requirements) framework [2] refines and extends the i∗ framework. The NFR framework is a process-oriented approach, in which quality requirements are treated as goals (called softgoals) to be achieved. The goals are derived from the stakeholders' needs and used as guidance while considering different design alternatives, analyzing tradeoffs and rationalizing various design decisions. A softgoal interdependency graph is used to support the goal oriented process of architecture design.

CBSP [3] is another process oriented approach; it defines five steps starting from taking a requirement under consideration and finishing with making trade-off choices regarding architectural elements and styles. Each requirement is assessed for its relevance to the system architecture components and connectors (buses), to the topology of the system or a particular sub-system, and to their properties. The intermediate CBSP model is used as a bridge while refining and transforming requirements to architectural elements such as components and connectors.

QASAR (Quality Attribute oriented Software ARchitecture design method) [9] is a method consisting of two iterative processes: the inner iteration includes the activities of software architecture design, assessment and transformation to quality requirements, whereas the outer iteration refers to a requirements selection process to be performed within the inner iteration. The approach is process oriented starting from functionality and adapting it to quality requirements. Thus, in this method quality requirements are not considered a driving force in architecture development.

2.1.3. Modeling product family architectures

There are several development methods that support family architecture development in different ways [10]. The COPA (Component-Oriented Platform Architecting) method starts from customer needs, stakeholder expectations, facts, existing architectures and from an intuition of a new architecture [11]. The views of the method, namely CAFCR (Customer, Application, Functional, Conceptual, Realization), assist in transforming requirements to architecture. However, the method does not consider quality requirements as architectural drivers of a product family.

The common properties of the PFA methods, e.g. FAST (Family Oriented Abstraction, Specification and Translation) [12], FORM (Feature-Oriented Reuse Method) [13] and KobrA (Komponentenbasierte Anwendungsentwicklung) [14], are that (1) they start scoping a product family from known facts (i.e. existing systems), (2) they lack the knowledge of how to transform requirements to architecture, and (3) they concentrate on functionality instead of quality.

2.2. Quality driven architecture development

Our earlier work forms the core of the Quality-driven Architecture Design and quality Analysis methodology (QADA®[1]). QADA is a quality-driven architecture development approach with a stakeholder-based definition of

[1] Registered trademark of VTT Technical Research Centre of Finland, http://virtual.vtt.fi/qada/.

architectural viewpoints and a set of predefined views, which use a set of diagrams for representing a view of architecture [15–17]. QADA contributes to software family engineering by providing a method for selecting an appropriate family architecture approach, a method for quality-driven architecture design, a method for evaluating the maturity and quality of the family architecture, and a technique for representing variation points in the family architecture. The QRF method extends QADA by providing a systematic method for defining quality requirements and transforming them to architectural models.

The main purpose of architecture is to [15,17,18]:

- provide an overview of software structure and its components,
- classify components into generic and specific categories,
- describe the responsibilities and contexts of components, and
- consider the appropriateness of architecture, i.e. the balance of business and technical issues.

Essentially, architecture provides a means of communication for reasoning and prioritizing quality attributes, and evaluating how quality requirements are met. Architecture is also essential in the management of development team members. It further provides for allocating and establishing work division, mapping responsibilities to services/components and vice versa, mapping functional and quality requirements to components/services, and clustering the components to be developed.

An architectural view is a representation of a whole system from the perspective of a related set of concerns [18]. In the literature, there are several approaches to the design of software architecture, concentrating on different views of architecture. However, among these approaches no general agreement can be found on a common set of views or ways to describe architecture. This disagreement arises from the fact that the need for different architectural views is dependent on, at least, three issues: system size, domain and stakeholders. System size and domain have an impact on the amount of stakeholders to be considered.

Service architecture is a set of concepts and principles for the specification, design, implementation and management of software services [19]. Software architecture of distributed systems is typically divided into thee layers; system infrastructure services, middleware, and applications. Service architecture embodies applications and middleware and is based on the widely accepted assumption and consensus that the wireless and mobile access systems will be converged with Internet systems. Maturing service technologies are also extending the global software market for generic middleware services.

Our focus is on the service architectures and the viewpoints needed in modeling service architectures. Thus, in QADA, for the two levels of abstraction – conceptual and concrete – four viewpoints are provided: structural, behavior, deployment and development. These viewpoints embody the quality of service architecture and that of the service using it. Qualities are visible at the architectural level only through the architecture documentation, i.e., in the views, models and diagrams and the notation used in these, as well as in the reasons behind the design decisions.

3. The QRF method

The QRF method consists of five steps:

1. Impact analysis,
2. Quality analysis,
3. Variability analysis,
4. Hierarchical domain analysis, and
5. Quality representations.

Fig. 1 presents an overview of the QRF method defined by a UML2 activity diagram. The swim-lanes are named according to the main categories of the stakeholders involved in applying the method. Business stakeholders are responsible for the market scoping activity of the impact analysis, which is the first step of the QRF method. The main output of this activity includes a list of stakeholders and the quality goals of the product family development. Concurrently with the market scoping, the domain stakeholders are defining the scope of the product family by using their knowledge of the domains related to the existing and emerging products. The output of this activity is a specified set of family assets used as input for the next two steps, quality analysis and variability analysis, carried out concurrently in an iterative and incremental way. The focus of the quality analysis is on separation of concerns; each quality requirement is specified by taking into account its relations to the quality goals, stakeholders, and product types. The goal of the variability analysis is to separate commonalities and variabilities by using the family assets as a starting point and making refinements based on the intermediate results of the quality analysis. In many cases, quality and variability are intertwined in product families; the quality of a product family depends on how well variability is managed, and the quality of product family members varies due to varying market requirements. In the next step, the quality requirements are mapped to functional capabilities of the product family by using a hierarchical domain analysis. This phase results in a taxonomy of services, recommendations and options for variability management, and in a definition of the architectural drivers. Finally, the quality requirements are represented in architectural models by addressing the defined architectural drivers and the views that are best suited for representing the most important qualities.

3.1. Impact analysis

The purpose of the first step of the QRF method is to define the scope of a product family, especially concerning its quality goals. The activities of this step are to categorize

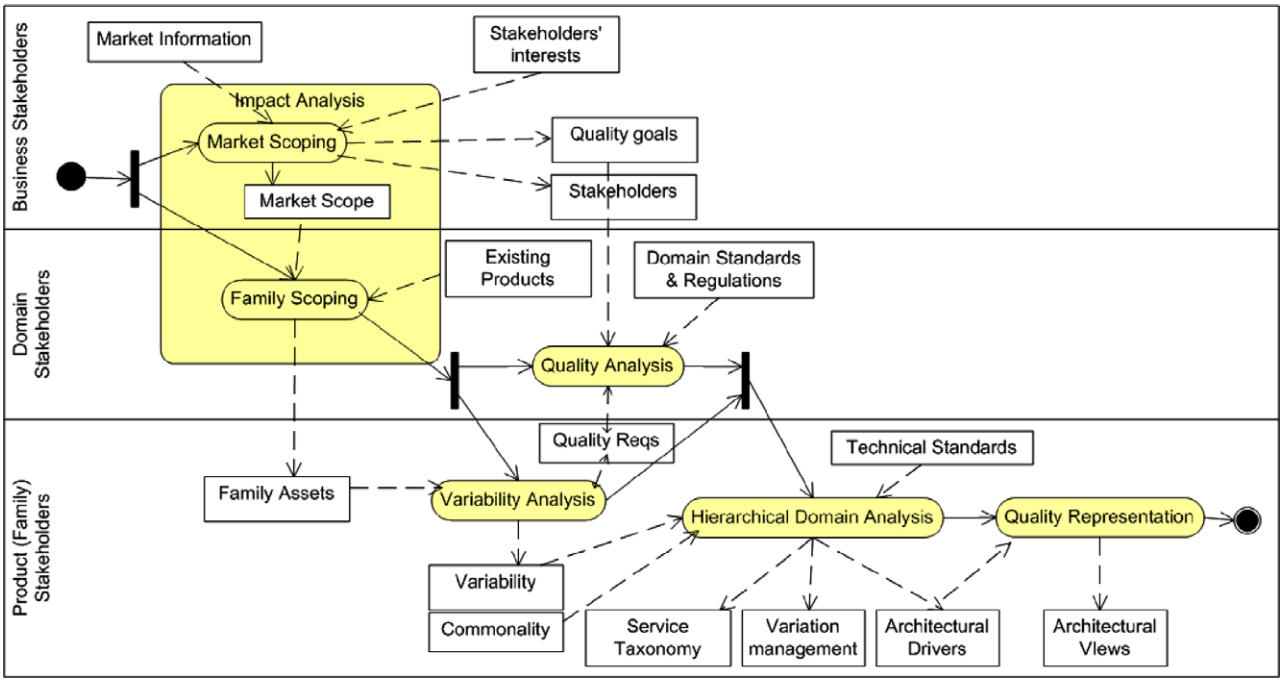

Fig. 1. The main steps of the QRF method.

the stakeholders of the system family from the business and development points of view and to define what to examine in quality evaluation and its rationale from the business and technology points of view. Therefore, the first activity of the impact analysis examines how business stakeholders' needs and markets affect the scope of a product family by identifying the external quality requirements for a product family. Thereafter, an internal product family scoping is carried out, focusing on the identification, evaluation and optimization of the entities, i.e. products, domains and assets, that should be included in the product family and represented in the respective software family architecture. Both business stakeholders (referring to customers, system users and company business managers) and domain stakeholders (i.e. domain experts) are involved in these activities.

Traditionally, product family scoping is started by selecting one key product around which the product family is established [20–22]. However, in service architectures business issues and quality requirements play an important role and therefore, the family scoping is started by identifying and defining the quality goals, not functionality.

In market scoping, customer needs and market forecasts are examined and clustered to groups of product family requirements. Data is also collected by interviewing business stakeholders. The collected data and market forecasts are examined for predicting product family requirements. Market segments define the business domains that are to be covered by the product family. The purpose is to set the market scope of the product family, to identify new potential in the markets and to define special requirements for the extensibility and changeability of product families for different business domains.

Consequently, the goal of product family scoping is to make decisions on which parts of the existing products are to be included in the family architecture. However, in order to make the decisions, the family architect has to have a clear understanding of why the capabilities identified in products, domains and assets are important for the family of products. Therefore, the knowledge of domain experts is essential in defining the boundaries of the product family.

Concerning quality requirements, the first step of the QRF method provides a list of stakeholders and their concerns that have to be supported by the product family architecture, a list of quality goals that have to be fulfilled by the product family architecture, and a list of potential family assets appropriate for the product family.

3.2. Quality analysis

The purpose of the second step is to separate the quality concerns related to business, constraints and functionality and to express the quality requirements in a way that they can later be traced and measured. Nowadays, quality analysis is made in an ad hoc manner; in practice quality requirements are not traceable or measurable. One reason is that the quality requirements of a product family have to be collected from various sources, which is laborious. Domain experts representing the developer organization, the customer organization or both, are interviewed for gaining tacit knowledge about the special requirements of the domain or domains related to the quality requirements and execution environments. Service Level Agreements may also be used to identify the quality requirements of a specific service domain.

Domain standards and regulations set requirements and constraints that have to be considered in a specific domain. Standards can be international, i.e. applied all over the world or in some continent, or national regulations, which may cause variation not only in functionality but also in quality. Furthermore, production stakeholders, such as product managers, are also interviewed in order to collect knowledge about the issues related to the product development organization, e.g. third party component/software providers and production facilities.

This step includes several activities. First, the quality requirements of the product family are visualized from customers' and developers' points of view. Second, the dependencies of quality on business are identified and the stakeholders and speciality of domains are defined. Third, the quality requirements are categorized based on the graphical representation of the dependencies between stakeholders, domains and required qualities, and finally, the quality categories are prioritized. For traceability, each quality requirement (QR) has an identification number, a description, stakeholder(s) interested in it, related business domains/customers, and related functional properties.

3.3. Variability analysis

The purpose of this step is to define the requirements that vary between the business domains and stakeholders, and to separate the commonalities and variabilities in the given domains. The main activities of this step include identifying the variation in quality and defining the variation dependency of quality requirements on the business domains and different stakeholders. Product family assets, a list of stakeholders and visualized QRs are used as input of this step. Variability analysis starts from business stakeholders' needs, and the dependencies of these needs on domains and other stakeholders. In addition, the commonalities and specialities of variations are identified, specified and linked to the stakeholders. Each domain and stakeholder requires functional and quality properties that may vary. Finally, the needs of production stakeholders, i.e. family architects, product architects, software component developers, and the like, are considered because they may also cause variation in functional and quality properties. However, only those properties that are visible to the users of the systems or products are considered by answering why these properties are required.

Evolution qualities, e.g. integrability and maintainability, are, typically, among the interests of the development organization(s), and therefore there is less variation in evolution qualities between product members. The execution qualities, e.g. reliability and performance, are usually different for each product member. In both cases, it is important to prioritize the requirements regarding their importance and impact on family members. Thus, the third step of the QRF method results in a categorized list of quality requirements including

- the importance of each QR,
- the variation of each QR, and
- the relation of each variable QR to the product members.

3.4. Hierarchical domain analysis

The purpose of this step is to bring together the information from steps 1 to 3 and to provide the information required for modeling and evaluating the architecture. Hierarchical domain analysis uses the clustering technique for grouping functional requirements to service categories, domains, sub-domains and functional and quality properties defined as responsibilities of services (cf. wireless service architecture in [17]). Thus, the hierarchical domain analysis results in a hierarchy of service domains and services, called a service taxonomy, and provides quality properties and their variation for the service domains, and finally, qualities for each service.

This step includes three main activities. First, the technical standards and existing knowledge of product developers are analyzed and utilized in categorizing the service domains and identifying the properties of the services. The identified properties of services determine the collection of categorized capabilities to be covered by the family architecture.

Second, the service categories, their capabilities and the quality requirements are combined. This is based on the service categories (resulting from the first step of hierarchical domain analysis) and the list of the prioritized quality requirements (resulting from the quality and variability analysis). This step also combines the variation between family members and services and defines the means of managing variations.

Third, the business drivers, first identified as key issues in the visions and strategic plans by interviewing business managers, are transformed to the drivers of the family architecture. Technical key issues, identified in an exhaustive hierarchical domain analysis, are also highlighted as technical, common or product specific drivers of the family architecture. These drivers form the architectural drivers of the first version of a conceptual product family architecture.

The actual family architecture is defined iteratively, while the involvement of product family stakeholders and product derivation staff is needed in the evaluation and validation of the concepts of family architecture.

3.5. Quality representation

In this step, quality requirements are represented in the architecture by means of a set of views. The step includes two activities; selecting the styles and patterns and describing specific, qualitative constraints.

Since styles and patterns support different qualities [17,23], their selection is based on the importance of the QR for the family architecture and on the support that each

style and pattern provides for this specific QR. Our approach provides a stylebase [23] from which the architect can get information about the quality properties of each style and pattern for making selections. For evolution QRs, the use of appropriate styles and patterns is the only way of achieving the desired quality.

Execution QRs, e.g. reliability and availability, have to be represented in the architectural models by adding quality requirements to the elements of the architectural views. This is made by using the quality profiles especially developed for defining reliability and availability QRs, including the identification numbers and numerical values for different dimensions, such as failure rate, fault treatment, and data correctness. The identification numbers and numerical values provide traceability and measurability for QRs, which are required in architecture evaluation.

For the rest of the process, the modeling of product family architecture follows the QADA methodology, as defined in [15–17,24].

4. Applying the QRF method

This section presents how the QRF method was applied to the DiSeP case example. The example is mainly based on applying the QRF method to evaluating integrability and extensibility (i.e. evolution qualities) from architectural models. Integrability is the ability to make separately developed components of a system to work correctly together. Extensibility is the ability to extend a software system with new features/services/components without loss of functionality or qualities specified as requirements. Reliability and availability (RA) (i.e. execution qualities) are used as examples of desired qualities in Sections 4.4 and 4.6 to illustrate that qualities may require different techniques in some steps of the QRF method. In overall, however, the steps of the method are the same for both evolution and execution qualities.

4.1. Case description

DiSeP is a distribution platform (Fig. 2) for software systems families formed in a networked environment. The services of the platform are mobile, enabling spontaneous networking. The hardware of the system consists of distributed computing units. Each computing unit, i.e. deployment node, is a platform for various services. The computing units join the network spontaneously by listening to the network and registering themselves to the network by using system services. After registration, all the registered services of the network are available.

The family architecture is service oriented, i.e. each product is composed of a set of services provided by DiSeP. The scope of the family is limited to the platform services so that the applications are considered only in the application interfaces provided on the top of platform services. Thus, the DiSeP platform embodies a layered service architecture.

The combination of services in deployment nodes may vary. Platform services can be mandatory, alternative or optional. The platform consists of services representing four different domains: *system service user interface, system services, basic services* and *communication services*. The applications have access to the platform services through system service user interfaces. System services provide services that are not autonomous but activated by the autonomous parts of the platform. System services are mandatory for each node, but they are active only in one node at a time. The services of the other nodes in the network use the system services of the active system services node. Basic services consist of controlling services, data management services and location services. Communication services provide messaging services for handling the communication between different units.

The DiSeP platform is intended to be utilized as a generic distribution platform for a variety of products. According to the service categories introduced in [17], we defined three end-user services, in which the platform was to be used. The end-user services domain consists of five service categories: mobile entertainment, mobile information, mobile communications, mobile commerce and critical services. We selected entertainment services from the mobile entertainment category, healthcare applications from the mobile information category and emergency services form the critical services category. These categories were selected because the products of each service category had their own separate quality requirements.

4.2. Impact analysis

The stakeholders of the DiSeP family that are interested in integrability and extensibility are:

- Application developers producing end-user services to be run on top of the DiSeP platform;
- Service and component providers offering their products for the use of the platform providers;
- Integrators using the platform as part of products as such. The assumption is that the platform can be sold to the integrators of the same company or other companies.

The developer stakeholders, i.e. software family architects, product architects, component designers and product maintainers, and the like, set slightly different requirements for integrability and extensibility, but due to reasons of simplicity, these are integrated into the integrator's viewpoint here.

The quality goals were defined for each relevant quality attribute, as illustrated in the fragment of the list of goals for integrability and extensibility (Table 1). The goals were justified by design rationale in order to explicitly show why each quality was required. Clustering requirements according to the stakeholders helped in tracing the owner of each integrability and extensibility (IE) requirement.

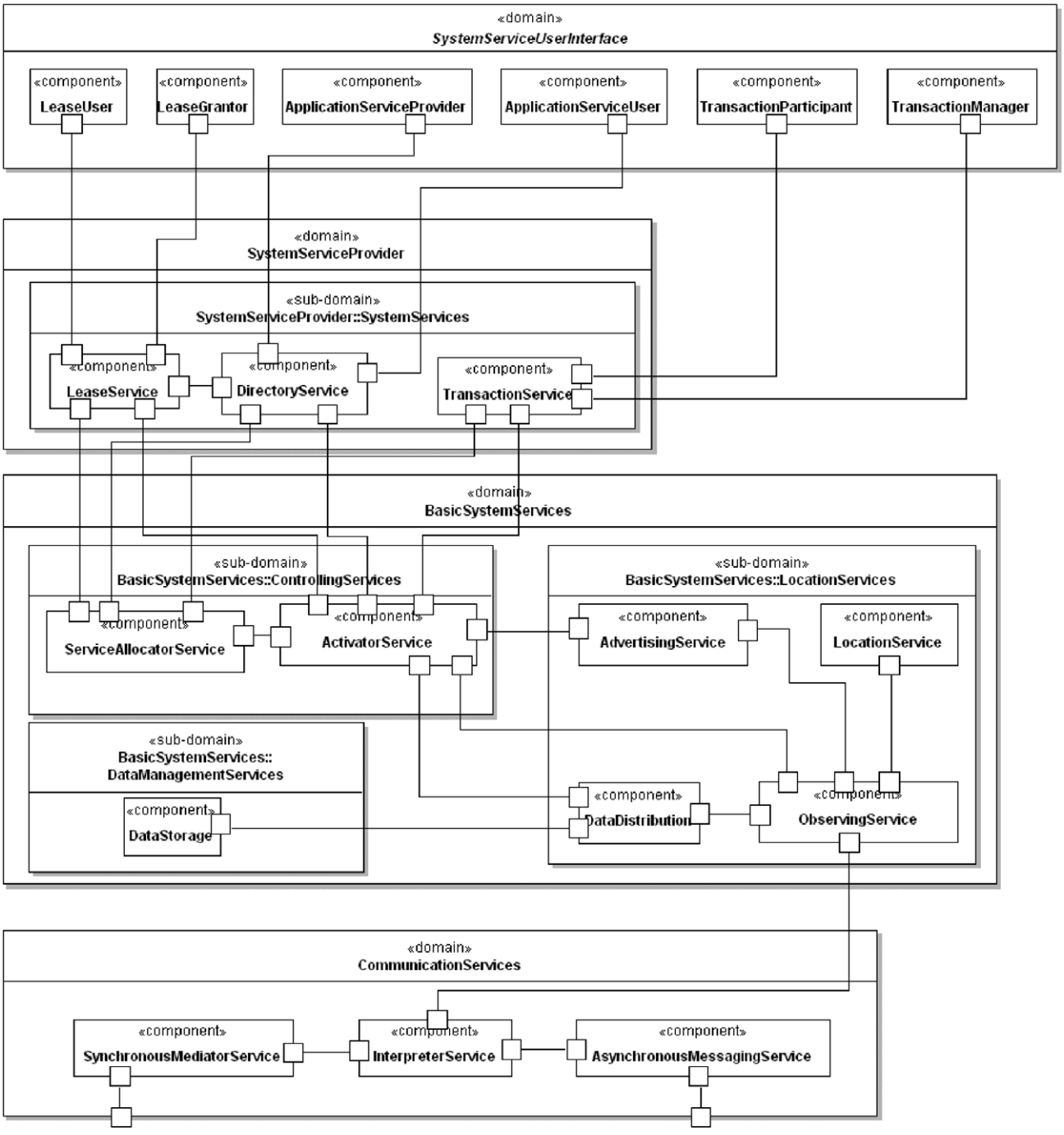

Fig. 2. Conceptual structural view of the DiSeP.

The main contributions of the first step of the QRF method are its support for identifying and defining the quality goals and the means of mapping the defined quality goals to the stakeholders. Explicitly defined quality goals help in understanding stakeholders' targets of interest. The rationale provided for each goal assists in understanding the business driver(s) behind the technology development. The existing methods omit business stakeholders and their interests in architecture development.

4.3. Quality analysis

After setting the quality goals, the quality requirements were further refined based on the knowledge gained from domain experts, standards, regulations, and customer expectations.

First, the i* framework was used to define the requirements from the viewpoints of end-users, application developers, component developers, product architects and family architects. Due to the limitation of the graphical

Table 1
Quality goals

Stakeholder	Integrability	Extensibility
Application developer	*Goal*: End-user services have to be integrated into platform user interfaces as third party components, i.e. a suitable API has to be provided *Rationale*: If applications are widely supported, the platform is useful for different kinds of products	*Goal:* New applications can easily be added on top of the platform *Rationale*: The main purpose of the platform
Service/component provider	*Goal:* Some services, e.g. data storage and protocols, can be acquired from third parties *Rationale*: The company wishes to focus only on the core functionality of DiSeP	*Goal*: Functionality of a component can be provided by a set of available COTS components *Rationale:* Customers can select technology they are familiar with/they trust
Integrator	*Goal*: Third party components have to conform with the styles of the product family architecture *Rationale*: The aim is to keep the product family architecture stable and evolving *Goal*: Different technology platforms are used in networked systems, in which these platforms shall be integrated regardless of the implementation languages and component models *Rationale*: The use of legacy systems and renewing of a networked system shall be possible	*Goal:* A new feature providing a new usage scenario can be added by minimal effort *Rationale:* A new business/usage scenario is required for each new application *Goal*: An existing service can easily be substituted by a third party service that provides additional functionality for new product types *Rationale*: While markets are changing, the service may become a commodity and be out of the scope of the company business

representation, the i∗ framework was adapted by removing everything but the most important qualities and responsible persons for them. By the revised i∗ framework, we could graphically represent which IE qualities were regarded as the most important and who the 'owners' of these qualities were. Thereafter, the IE requirements were more thoroughly defined by grouping them into qualities related to architecture, components/services, and applications, as depicted in a fragment of the requirement list in Table 2. Categorized QRs helped to trace the responsible persons of qualities; architects, designers, and application developers. IE qualities are mostly related to business and development. Some of them may be expensive to realize and trade-offs have to be made. Therefore, the relations to other qualities, constraints and functionalities have to be specified.

The main contributions of the second step of the QRF method are the graphical representation of the quality requirements of the most importance and their interest groups, scoping of the quality requirement to architecture, components and applications, and defining how the quality requirements relate to business and other capabilities of the product family. The quality requirements of most importance form the key drivers of architecture development and their fulfillment provides the biggest benefit. Thus, they are the main interest of a product family architect. Traceability of quality requirements to their 'owners' assists in making tradeoffs between cross-cutting quality requirements while seeking a balanced and optimal architecture. There is a gap between existing requirements engineering methods and architecture design methods; the QRF method tries to close this gap.

Table 2
Refined quality requirements

Category	ID	QR description	Interest groups	Business domains/ customers	Related to
Architecture	I1	Style conformance	Family architects, product architects	All domains	All services
	E1	Extensibility of service interfaces	Architects, component developers	Emergency services	System services
Component/Service	I4	Interoperability of networked services	Architects, component/ service providers	All domains	Basic services
	E4	A new feature can be added to the system and basic services	Product architect	Emergency and healthcare services	The whole middleware: application platforms, system services and basic services
Application	I7	Easy integration of applications	Content/application providers	All domains	System service user interface
	E5	New applications are easily added	Product architect, application service providers	Entertainment services	System service user interface

4.4. Variability analysis

There may be three kinds of variation in quality between the members of a family. First, there can be variability among different quality attributes. Second, there may be different levels in quality attributes. The levels define how critical the requirements for certain quality attributes are for the specific product. Third, functional variability may indirectly cause variation in qualities, and vice versa.

Only one variable IE requirement was identified: Variable service user interfaces were required in the different nodes of the networked systems. Consequently, the service user interfaces had to allow configuring during installation and at run time.

Concerning execution qualities, several variable RA requirements were identified due to the different levels of reliability and availability. For example, for emergency services, a high degree of service availability and recovery were required, whereas for entertainment services, the service availability only needed to be of medium rate. Furthermore, a controlling and monitoring unit of some kind was required in the context of emergency services to ensure a failure-free service execution, whereas for entertainment services it was not required. Since reliability and availability are execution qualities, the variable requirements involve both structural and behavioral aspects of architecture. Fig. 3 represents a Strategic Dependency model of the i* framework that describes the variability of RA requirements between three product family members [25]. The circles in the i* framework correspond to stakeholders, rectangles to the required functionality and ellipses to the RA requirements. The arrows describe the dependencies. The family specificity is highlighted in grey.

The third step of the QRF method contributes in identifying quality variability and points of time when variation takes place. The existing variability management techniques address functional variability, not quality variability. To the aim of managing quality variability, it is essential to identify why, what and when quality variability is required. The defined three quality variation types help in identifying quality variation. When quality variation is identified, it is possible to create a tactic (or to select one of the existing tactics) for managing quality variability. Identifying when the variation takes place allows the architect to define binding time and select appropriate mechanisms for it.

4.5. Hierarchical domain analysis

When the IE quality categories had been defined and their prioritization done, the hierarchical domain analysis was used to cluster the functional requirements. The functional requirements were thought of as responsibilities of the family members. The responsibilities were mapped to

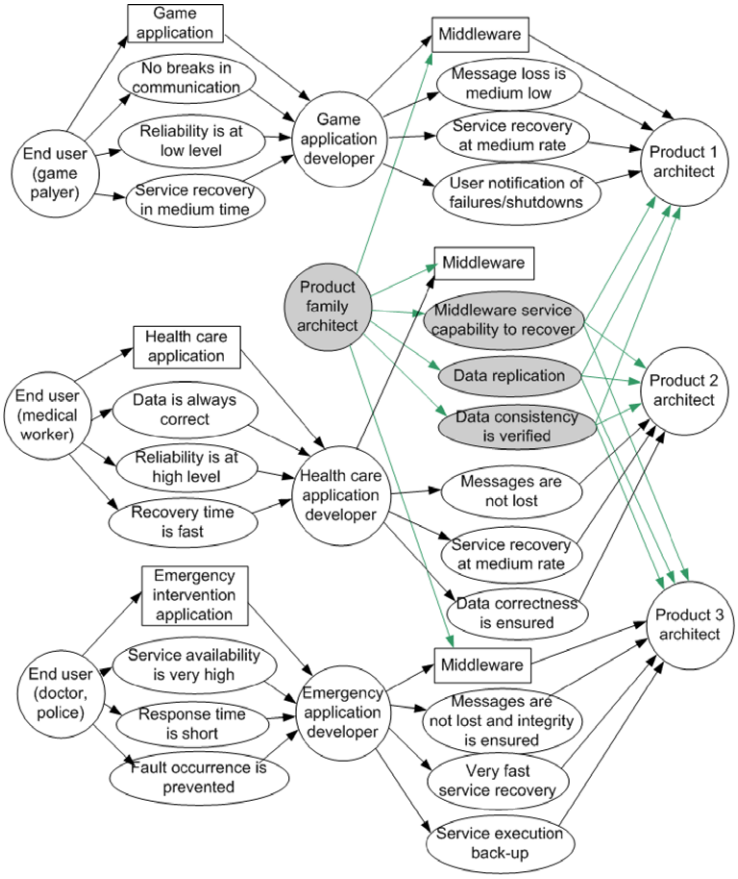

Fig. 3. Variability in RA requirements between product family members.

the categorized services and the properties of services were defined as capabilities organized as domains, sub-domains, components or services of a component. A hierarchical list was used as a tool in grouping functional capabilities into domains and services because a textual list was more convenient than a graphical representation while making changes and restructuring the capabilities of the product family. When a desired grouping had been achieved and the responsibilities of services defined, the quality requirements were mapped to each service category and each service.

Table 3 presents the DiSeP service categories, the required IE qualities and their scope. Furthermore, the architectural quality drivers were defined based on the importance of the IE requirements (high/medium/low) and the scope of their impact by specifying the seriousness of conflict if the quality requirements were not met. In the case, if the QRs with high importance and broad impact are not met, the quality goal of the family architecture is broken. In DiSeP, style conformance (I1), diversity of languages and component models (I2), substitutability of middleware services (I3) and component extensibility (E3: new components added at run-time, E4: new features added to existing services) were ranked on top of the QR priority list. Thus, these QRs were considered the quality drivers of the product family architecture.

The main contribution of the fourth step of the QRF method is a 'standard' service taxonomy, explicit mapping of quality requirements to services, and domain based clustering. The service taxonomy provides a reusable vehicle for defining and categorizing the functional capabilities to services. The explicit mapping of quality requirements assures that quality requirements are considered in architecture design. Domain based clustering helps in assigning work to the most appropriate persons, i.e. architects, quality experts, implementation experts, etc. To our knowledge, none of the existing software or service engineering methods provides this kind of support for mapping functional and quality capabilities to architecture.

4.6. Quality representation

IE qualities were represented in architectural models by using design patterns, e.g. adapter, wrapper, and façade, which assisted in integrating components/services differing

Table 3
IE qualities mapped to hierarchical service categories

Domain	Sub-domain	Service	Purpose/responsibility	IE requirement
Service user interface		Application service user	Access to application services through the directory	E5, I7, I8
		Application service provider	Enables the user to create, register and un-register a service	
		Lease user	Lease (re)negotiation	
		Lease grantor	Leases granting	
		Transaction manager	Request a transaction	
		Transaction participant	Participation in a transaction	
System services		Lease service	Management of leases	I1-I3: all services I4, I5: system and basic services E1: system and basic services E2, E3: system services E4: all services
		Directory service (DS)	DS interface to distributed data storage	
		Transaction service	Performs and tracks transactions	
Basic services	Controlling services	Activator service	Controls services	
		Allocator service	Activates services according to received requests	
	Data management services	Data storage	Permanent data base	
	Location services	Data distribution	Assists in DB storage, tracks needed redundancy, negotiation about copies, transfers, and deletions	
		Location service	Manages location information, service registration, tracks location maps, announces of available system services	
		Observing service	Routes messages from communication services to appropriate services	
		Advertising service	Locates the system service provider, announces available services	
Communication services	Messaging service	Interpreter	Encodes/decodes XML messages	I3: protocol components I5: all communication services
		Asynchronous messaging	Creates and manages mailboxes; Receives, hosts and notifies messages	
		Synchronous messaging	Sends synchronous messages through defined protocols, and transforms received services to the interpreter	

semantically or syntactically from the style of the family architecture. The adapter pattern was used to harmonize the COTS, OS and in-house databases behind the same interface and to allow different protocols to be used in communication (Fig. 4). A wrapper was used for adapting open source components to the syntax of the interfaces specified by the family architecture. The façade pattern provided an extension point for variable application interfaces. The decorator pattern was applied to harmonize the diversity of component models by decorating other component models with the required interfaces of the chosen component model.

RA requirements typically lead to certain structures or functionalities. Therefore, variations in RA requirements between family members may result in different design decisions concerning the architectural style, components and component collaborations. For example, the required high degree of availability and recovery of the emergency service could be achieved by using an architectural solution that enabled back-up service execution. The medium rate service recovery capacity of the entertainment application could be ensured by modifying components to implement, for example, a recovery mechanism. Thus, the variable RA requirements affect the whole architecture design of the different family members. The cost and effort of design is typically higher in the case of high RA level products.

RA qualities were represented in architectural models using the defined RA profiles. These profiles are UML extension mechanisms [26], which can be used for modeling certain quality aspects in architectural models. The abstraction levels of QADA enabled a separation of required and provided profiles. The required profile corresponded to RA requirements, i.e. what the system was required to support, whereas the provided profile corresponded to the implemented RA. The profiles consisted of RA dimensions and values. The dimensions, e.g. probability of failure, error detection and redundancy, were needed for representing the metrics and means for the RA aspects in architecture.

The RA properties were first mapped to RA dimensions attached to architectural elements, such as components and connectors. For improved visibility, the RA properties can be represented in architecture using notes. In Fig. 5, the RA requirements with gray notes are the common RA requirements of the product family. Other RA requirements are system specific, indicated by identification numbers, e.g. R1-S3 means reliability requirement #1 for system #3 (emergency applications).

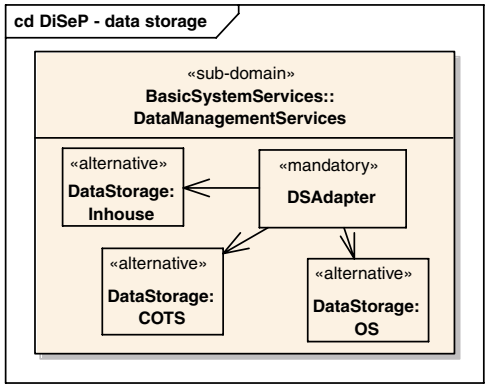

Fig. 4. Various DBs adapted to the product family architecture.

Quality representation in architectural models is essential in order to evaluate quality issues at the architecture level. The Q-Stylebase [23] is a repository of architectural styles and patterns used as an architectural knowledge base while designing and evaluating architecture. Styles and patterns support especially evolution qualities, such as integrability and extensibility. However, for execution qualities like reliability there are only a few styles (e.g. Simplex ABAS, Implicit invocation, Blackboard) to select from. Different approaches for representing quality aspects in architectural models are required for execution and evolution qualities because the evaluation techniques used for evaluating execution and evolution qualities differ; execution qualities are evaluated by quantitative and qualitative methods [25], and evolution qualities are measured by using qualitative scenario-based methods [27]. Therefore, the type of quality representation in the architectural models depends on what kind of information is required for evaluation.

5. Lessons learned

In this section, the observations made while applying the QRF method to the laboratory experiment and an industrial case of product family initiation are presented.

5.1. Eliciting quality requirements

While eliciting quality requirements for the laboratory experiment, the literature of the similar kinds of experiments and the expertise of senior researchers were used as a starting point. Although there were many good examples available in the research field, none of them really explained how to define quality properties. This was the case especially with the execution qualities of reliability and availability. The most important source for defining reliability requirements was provided by the interviews of scientists who had worked many years in the area of safety critical systems. The definition of evolution qualities was easier because of our own experience of software reuse and product family engineering.

In the industrial experiment, although there were no lack of knowledge, it was scattered very broadly within the organization and among its customers. Therefore, a great number of interviews had to be conducted, including business managers of the organization, representatives of six customer groups of the product family and some external parties. There was also a lot of documentation available but it mostly concerned the solution domain, not the problem domain or its requirements. Based on the reviewed documentation, it was not clear how to define quality properties, and therefore, QRs were typically improperly defined or

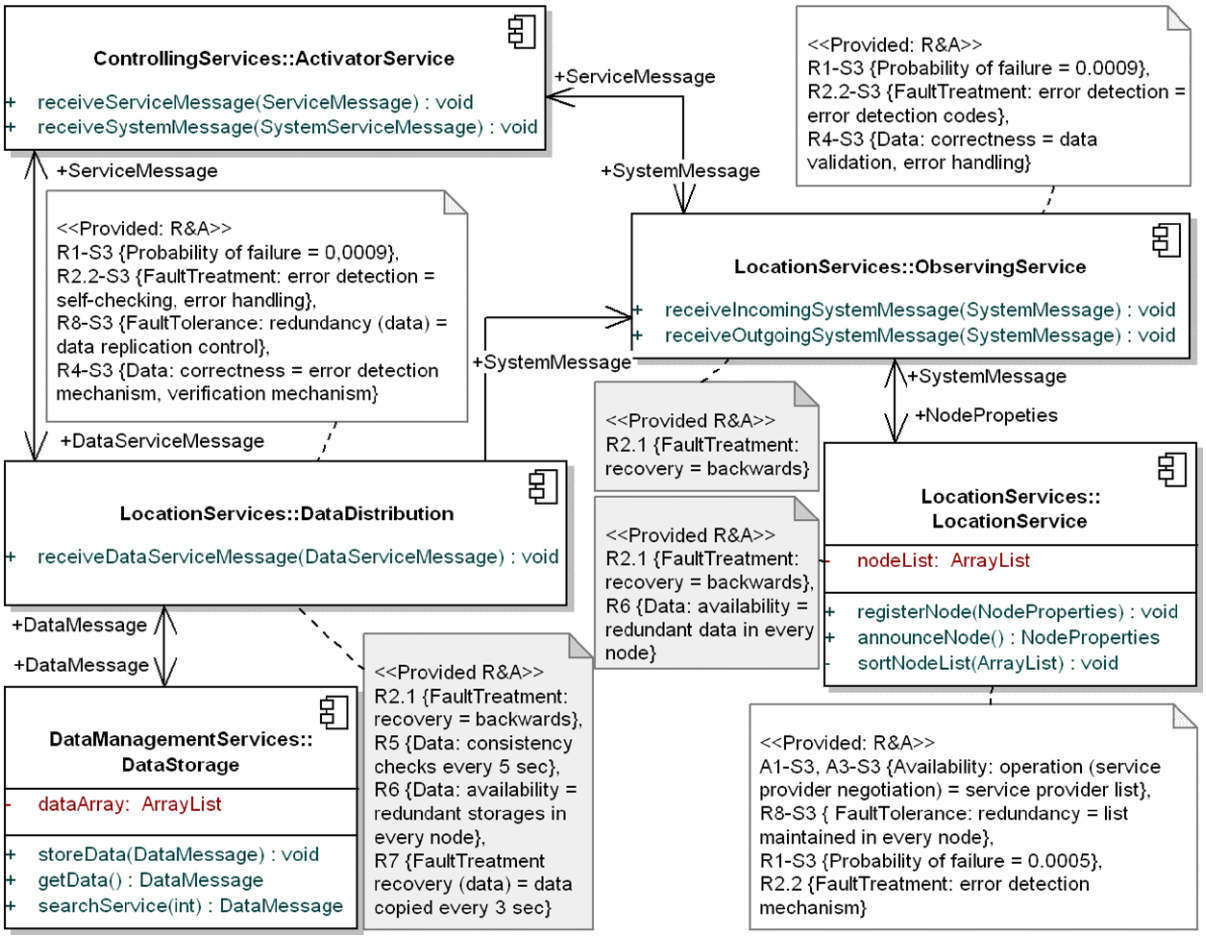

Fig. 5. Concrete structural view of the DiSeP with RA properties.

omitted in requirements specifications. One reason was the lack of time; collecting quality requirements involves using a number of sources, thus requiring a lot of time for conducting interviews, reading documents, and eliciting knowledge of domain experts and software developers. In the industrial experiment, seven weeks were used for gathering and defining approximately 80 requirements, while further refinements were still required afterwards. Moreover, in the specifications of existing systems, while many quality requirements were not mentioned, they were still considered the default properties of the systems in that domain (by domain experts). Furthermore, when a QR was defined, its proper meaning was often difficult to understand without domain expertise. Workshops provided a successful means of clarifying the meaning of QRs and achieving a common understanding among the development staff of the product family. As a concrete result, a documentation template for the requirements specification was defined.

In summary, quality definitions are more often led by standards than by the needs of customer groups. One reason for this is that product family engineering is traditionally based on existing technologies and products, and not on business requirements, which is the most recent trend in service architecture development.

5.2. Managing variable quality requirements

In the laboratory experiment, defining variation in evolution qualities was easy in itself but establishing the variation of execution qualities required clear understanding of the specifics of all family members and their impact on the functionality of the whole family. The i* framework helped in identifying and clarifying the variations and associations between the quality requirements of a product family. However, the representation was reduced so that only the most important QRs were illustrated in the same graph. The use of tables provided a tool for mapping stakeholders, QRs, domains and priorities to each other. The laboratory experiment included 27 QRs, of which nine were product family QRs and the rest product specific QRs. Thus, it can be justified that the use of the i* framework and tables provides appropriate tooling for managing variable quality requirements if the number of QRs is smaller than 30.

In the industrial experiment, the use of the i* framework turned out not to be an appropriate method because of the great number of functional and quality requirements, which were handled in a similar way. Therefore, the clustering technique was used. First, the customer groups and quality attributes were used as basis of the categorization.

Finally, these categories were refined into one category, which was common to all family members, and three other categories, which included specific QRs of these particular domains. Tables and a QR template were used as tools for managing variable QRs. However, variability management would be easier and less time consuming with suitable tool support. Therefore, it can be concluded that the management of variable QRs can be handled without specific tools while initiating a product family, but tooling is essential for illustrating QRs and their relationships and also for the evolution of the product family.

5.3. Transforming quality requirements to architecture

The use of hierarchical domain analysis worked well in both cases and the mapping of QRs was a straightforward activity when defining responsibilities (functional and quality) in an iterative way. Textual representation was preferred because graphs would have been too laborious due to the iterative nature of the work. Identifying the architectural drivers (i.e. quality drivers) of the product family helped in designing and evaluating the family architecture by keeping the focus on the issues of most importance. The architectural drivers also defined the most important qualities of the common parts of the software family. Focusing on the quality drivers was found to provide a remarkable return on investment because the common parts were used in each family member. Thus, addressing the quality of commonalities is likely to improve the quality of the product family.

5.4. Representing qualities in models

Documented styles and patterns provide support for achieving a high standard in evolution qualities. However, this kind of support is totally missing for execution qualities. Only one architectural pattern was especially targeted at reliability, and a couple of design patterns were recommended to be used in improving reliability. No support was offered by the modeling languages or tooling for representing execution qualities or their variation in architectural models. Therefore, we developed a stylebase as an extension of a commercial tool [23] and applied it to find appropriate design patterns for modeling integrability elements for architectural designs. For modeling RA requirements, UML was extended with specific RA profiles. However, these tools are still prototypes and thus not available to the software engineering community. Therefore, we are working on providing these tools for an extended range of use by offering them to the open source community as add-ons to Eclipse.

5.5. Incremental development and evolution of the QRF method

The QRF method was developed incrementally in three phases. The incremental development enabled the method to evolve and to be adapted as needed when using it in case examples.

First, the QRF method was applied to a laboratory experiment. The main result of this phase was the definition of the evolution and execution qualities for the DiSeP family. One of the main observations in the method development at this point was the need to define the IE and RA qualities in different ways. The prioritization of quality requirements and variability definition was another main goal of this phase.

Second, the method was applied to an industrial product family. In this phase, the method development concentrated on the definition of stakeholders, the trade-off analysis and the service taxonomy of that family. The main contribution of the phase was the selection of the architectural drivers. At this point, the other quality attributes were also taken into account, as well as the real variants.

Finally, the method was refined and reapplied to the DiSeP family. The purpose was to discover a common way to define quality requirements and to transform them to architectural design. The objective was to define architecture in a way that would allow the evaluation of qualities to be performed directly from the architectural models. Therefore, specific tools, i.e. RA profiles and the Q-Stylebase, were developed to help in representing qualities in architectural models and evaluating them at the architecture level.

Our most recent work focuses on two new topics; how quality variability should be considered in open source based software development and how quality variation should be modeled in order to manage quality variation at run-time.

6. Concluding remarks

Although there have been several attempts to fill the gap from requirements engineering to architecture modeling, there is no systematic approach available for defining and transforming quality requirements, including variable requirements, to the models of a product family architecture. The QRF method introduced in this paper is a systematic method designed for defining quality requirements, mapping the requirements to architectural models and for enabling quality evaluation at the early phase of product family development. The method defines steps and techniques for eliciting quality requirements, defining qualities in a meaningful way, and transforming and modeling quality properties in family architecture in such a way that allows the architecture to be evaluated against the quality goals derived from business drivers. The impact analysis defines the interested stakeholders and the quality goals concerning the quality of the family, whereas the quality analysis separates the quality concerns related to business, constraints and functionality. The variability analysis identifies the variability in qualities, while the hierarchical domain analysis brings together the required information for modeling and evaluating the architecture. Finally, in quality representation, the quality requirements are represented in architectural models.

The QRF method is suitable especially for product families where quality is a key issue. This is due to the broader impact of the existence or non-existence of specific quality attributes on all family members, and also to the level of management of variable qualities required for the diversity of market segments. The method has proven to work well in cases with relatively few QRs, while large product families will require additional convenient tools for recording, visualizing and managing quality requirements. Further research is also required especially for quality modeling and tooling, including

- standardized profiles for representing all execution qualities in UML models,
- extensible modeling platforms that can be configured according to the architects' preferences and the qualities of the highest importance, and
- automated tool support for evaluating QRs from models.

First of all, code generation from models shall be properly supported; a correct model should be able to transform to correct code, which still remains to be realized in practice.

A new research item related to service engineering was also identified in the course of the study: the need to be able to manage quality variability not only at development time but also at run-time. Finally, to be accepted by the service engineering community, the QRF method will require a uniform, formal and automated way of defining and managing quality variability.

References

[1] L. Chung, D. Gross, E. Yu, Architectural design to meet stakeholder requirements, in: The 1st Working IFIP Conference on Software Architecture, Kluwer Academic Publishers, San Antonio, TX, USA, 1999.

[2] L. Chung, B.A. Nixon, E. Yu, J. Mylopoulus, Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers, Boston, 2000.

[3] P. Gruenbacker, A. Egyed, N. Medvidovic, Reconciling software requirements and architectures with intermediate models, Software and Systems Modeling 3 (3) (2003) 235–253.

[4] OMG, MDA, Guide Version 1.0.1. omg/2003-16-1, J. Miller, J. Mukerji, (Eds.) 2003, Object Management Group. 62 p. Available from: http://www.omg.org/cgi-bin/doc?omg/03-06-01.

[5] IEEE, IEEE Standard for Software Quality Metrics Methodology, in: Std -1061-1998, Institute of Electrical and Electronics Engineers, New York, USA, 1998.

[6] ISO/IEC, Software engineering – Product quality, Part 1: Quality model, in ISO/IEC 9126-1:2001. 2001, International Organization of Standardization and International Electrotechnical Commission.

[7] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, Addison-Wesley, Reading, MA, USA, 1998.

[8] M. Matinlassi, E. Niemelä, The Impact of Maintainability on Component-based Software Systems, in: Euromicro 2003, Antalya, IEEE Computer Society, Turkey, pp. 25–32.

[9] J. Bosch, Design and Use of Software Architectures: Adopting and Evolving a Product-line Approach, Addison-Wesley, Harlow, 2000.

[10] M. Matinlassi, Comparison of software product line architecture design methods: COPA, FAST, FORM, KobrA and QADA, in: The 26th International Conference on Software Engineering (ICSE 2004). IEEE Computer Society, Edinburgh, UK, 2004, pp. 127–136.

[11] P. America, H. Obbink, R. van Ommering, F. van der Linden, CoPAM: A component-oriented platform architecting method family for product family engineering, in: P. Donohoe (Ed.), Proceedings of the First Software Product Lines Conference, Software Product Lines, Experience and Research Directions, Kluwer Academic Publishers, Boston, USA, 2000, pp. 167–180.

[12] D.M. Weiss, C.T.R. Lai, Software Product-Line Engineering: A Family Based Software Development Process, Addison-Wesley, Reading, MA, USA, 1999.

[13] K.C. Kang, J. Lee, P. Donohoe, Feature-oriented project line engineering, IEEE Software 19 (4) (2002) 58–65.

[14] C. Atkinson, J. Bayer, C. Bunse, E. Kamsties, O. Laitenberger, R. Laqua, D. Muthig, B. Paech, J. Wust, J. Zettel, Component-based Product Line Engineering with UML, Addison-Wesley, London, New York, 2002.

[15] E. Niemelä, M. Matinlassi, P. Lago, Architecture-centric approach to wireless service engineering, IEC Annual Review of Communications 56 (2003) 875–889.

[16] A. Purhonen, E. Niemelä, M. Matinlassi, Viewpoints of DSP software and service architectures, Journal of Systems and Software 69 (1-2) (2004) 57–73.

[17] E. Niemelä, J. Kalaoja, P. Lago, Toward an architectural knowledge base for wireless service Engineering, IEEE Transactions on Software Engineering 31 (5) (2005) 361–379.

[18] IEEE, IEEE Recommended Practice for Architectural Descriptions of Software-Intensive Systems, Std-1417-2000, Institute of Electrical and Electronics Engineers Inc., New York, 2000, 23 p.

[19] TINA, Service Architecture Specification, 1997. Available from: http:www.tinac.com.

[20] K. Schmid, A comprehensive product line scoping approach and its validation, in: The 24th International Conference on Software Engineering, ICSE, Orlando, FL, USA, ACM, 2002, pp. 593–603.

[21] K. Schmid, I. John. Starting product lines (I) – systematic product line planning and adoption, in: The Third Software Product Lines Conference, Boston, MA, 2004, 44 p.

[22] E. Niemelä, T. Ihme, Product line software engineering of embedded systems, ACM SIGSOFT Software Engineering Notes 26 (3) (2001) 118–125.

[23] J. Merilinna, E. Niemelä, A stylebase as a tool for modelling of quality-driven software architecture, in: Proceedings of the Estonian Academy of Sciences Engineering. Special issue on Programming Languages and Software Tools., vol. 11, No. 4, 2005, pp. 296–312.

[24] M. Matinlassi, E. Niemelä, L. Dobrica, Quality-driven architecture design and quality analysis method, A revolutionary initiation approach to a product line architecture, VTT Technical Research Centre of Finland, Espoo, 2002.

[25] A. Immonen, A method for predicting reliability and availability at the architecture level, in: T. Käkölä, J.C. Duenas (Eds.), Software Product Lines, Research Issues in Engineering and Management, Springer Verlag, NewYork, 2006, pp. 373–424.

[26] OMG, UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, Object Management Group, 2003.

[27] M. Matinlassi, Evaluating the portability and maintainability of software product family architecture: Terminal software case study, in: The 4th IEEE/IFIP conference on software architecture, WICSA 2004. IEEE Computer Society, Oslo, Norway, 2004, pp. 295–298.

Publication II

A service requirements engineering method for a digital service ecosystem

Service Oriented Computing and Applications,
Vol. 10, Issue 2, pp. 151–172.

SOCA (2016) 10:151–172
DOI 10.1007/s11761-015-0175-0

ORIGINAL RESEARCH PAPER

A service requirements engineering method for a digital service ecosystem

Anne Immonen · Eila Ovaska · Jarmo Kalaoja · Daniel Pakkala

Received: 3 September 2014 / Revised: 2 February 2015 / Accepted: 3 February 2015 / Published online: 14 February 2015

Abstract A digital service ecosystem enables value creation and the co-development of services in a value network under a common ecosystem regulation. The ecosystem members are able to focus on their core competences and can strengthen their forces by co-operating; yet remaining able to act independently. However, due to regulated environment, the ecosystem elements—i.e. ecosystem members, capabilities, infrastructure and the existing ecosystem assets—have an influence on digital service engineering, especially in the service requirements engineering phase. The main contribution of this paper is to describe how to specify the requirements of digital services in a digital service ecosystem. To this aim, this paper introduces the basic definitions and elements of the digital service ecosystem, and a scenario-based service requirement engineering (RE) method for the digital service ecosystem. A practical example is given to illustrate the use of the RE method. The collected feedback from the RE method users highlights the user experiences on the advantages and limitations of the proposed method.

Keywords Requirements engineering · Digital service ecosystem · Service innovation · Service co-creation

A. Immonen (✉) · E. Ovaska · J. Kalaoja · D. Pakkala
Digital Systems and Services, VTT Technical Research Centre of Finland, P. O. Box 1100, 90571 Oulu, Finland
e-mail: Anne.Immonen@vtt.fi

E. Ovaska
e-mail: Eila.Ovaska@vtt.fi

J. Kalaoja
e-mail: Jarmo.Kalaoja@vtt.fi

D. Pakkala
e-mail: Daniel.Pakkala@vtt.fi

1 Introduction

Recently, digital service providers have been strengthening their forces by co-operating, creating value networks flexibly and dynamically to provide services under the concept of a digital service ecosystem. A digital service ecosystem is a new kind of self-organised environment that addresses openness and dynamicity, enabling collaborative innovation and co-creation among ecosystem members. A digital service can be anything that is delivered digitally, is entirely automated and which is controlled by the customer of the service [1]. Service development in a digital service ecosystem sets new kinds of features for the service engineering process, but also new challenges. RE in a digital service ecosystem is not yet a standardised process, and only few studies exist. Like in service-oriented systems, service ecosystems include RE challenges, such as requirements change and evolution, quality requirements gathering and assessment, and uncertainties caused by the dynamic nature and unknown deployment environment, composition and users [2–7]. Moreover, digital service ecosystems provide new challenges in co-evolution among ecosystem members and in customer participation. Therefore, the models are required to propagate value in service value network, contextualise requirements, map them to sub-systems and communicate them to stakeholders [8,9].

The members of digital service ecosystem aim at the co-innovation and co-creation of new digital services within the dynamic value networks, while the utilisation of existing assets of the ecosystem assists in achieving the business goals. However, the current service engineering (SE) approaches do not define what the ecosystem elements are and how to go further from service innovation to service requirements specification. The SE approaches that utilise the existing assets, such as knowledge, do not consider the

ecosystem context. Thus, digital service ecosystems require a new kind of RE method that:

- Defines the ecosystem elements that are involved in service RE and defines the phases, activities and techniques to be used in each RE phase,
- Enables to define the role of each ecosystem member in the service RE process, supports the members' participation in all RE phases and provides the practices for co-innovation and co-creation,
- Helps in identifying the role of an ecosystem member in each value network depending on the context, and enables the value co-creation via digital service engineering in accordance of the roles and efforts,
- Assists in innovating new digital services by exploiting the existing resources (i.e. knowledge, assets and services) and maximising their use in different contexts, and
- Makes it easy to develop digital services that are interoperable, available and easily consumed by taking into account the specific capabilities of the ecosystem.

As a result of our work, this paper provides a service RE method for digital service ecosystems. The service RE method provides the following contributions:

- *Definitions* The digital service ecosystem is defined based on a thorough state-of-the-art analysis related to different kinds of ecosystems: business, service and software. Accurate definitions are required in order to get a mutual understanding of what digital service ecosystems embody.
- *Elements* The elements of the digital service ecosystem that influence the service RE have been defined. Definition of the elements that are present in the dynamic structure and behaviour of the digital service ecosystem makes it easier to communicate, negotiate and understand the big picture of a digital service ecosystem and its way of influencing service engineering.
- *Service innovation* The method enables the ecosystem members to innovate digital services by defining the scenarios and use cases that describe business goals and the usage of new digital services. Service innovation is supported by the existing ecosystem assets, such as the domain model, and the templates for requirements elicitation and identification, and for communication, knowledge sharing, negotiation and decision-making.
- *Business analysis* The method enables a multi-analysis approach that helps in making design decisions based on accurate justifications. The analysis provides insights into the business potential of new digital services by exploring market trends, customer needs and existing business know-how, also combining the impact analysis with the results of the analyses of risks, implementation technology and its complexity.
- *Requirement analysis, negotiation and specification* The method provides a repetitive activity-loop of service requirements analysis, negotiation and specification, where service ecosystem members are actively collaborating in defining a coherent and complete set of service requirements specifications. These specifications are provided as an output of the service RE method to the next phase of the service engineering—service architecture modelling.

This paper is organised in the following way: Sects. 2 and 3 provide a definition of the digital service ecosystem and the service engineering model as part of it. Section 3 also introduces the service requirements engineering method and practices in the context of a digital service ecosystem. A practical example that guides the use of the service RE method is provided in Sect. 4. Thereafter, in Sect. 5 the lessons learnt are provided, which helps users in adopting the service RE method by illustrating its strengths and shortcomings based on empirical evidences. In addition, Sect. 5 describes our ongoing research on applying the RE method, and future research directions. Section 6 summarises the main research results and closes the paper.

2 State-of-the-art

2.1 Business ecosystem, digital service ecosystem and software ecosystem: definitions

There are three different types of ecosystem definitions that are related to each other: the business ecosystem, the digital service ecosystem and the software ecosystem. The difference between these three is illustrated in Fig. 1. The digital service ecosystem can be positioned between the business and software ecosystem, taking characteristics from both sides and filling the gap between the two. Currently, as far as we know, there exists no research about digital service ecosystems. The business ecosystems [10–12] and software ecosystems [13–15] have been interested researchers extensively. Recently, also research has been carried out on service ecosystems [8,16,17]. Service ecosystems are closely related to research in the area of 'service value networks' and the 'Internet of services' [16]. Service value networks provide business value through the agile and market-based composition of complex services from a pool of complementary service modules by the use of ubiquitously accessible information technology [18]. The concept 'Internet of services' considers the Internet as a global platform for the retrieval, combination and utilisation of interoperable services [19]. Thus, especially in the case of web services, the

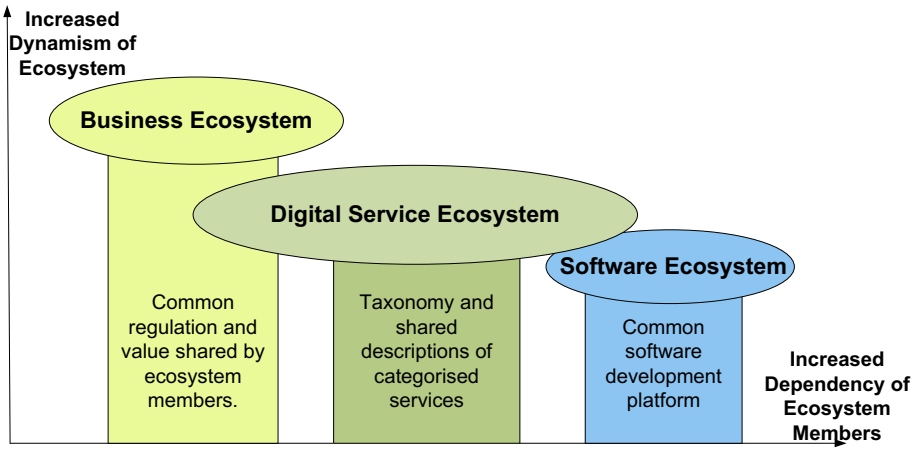

Fig. 1 Autonomy and dependencies of the different types of ecosystems

service ecosystem has gained a lot of attention in research [16,20,21].

The business ecosystem is a dynamic structure of organisations that work together in a specific core business [12], creating value in a network of actors. The ecosystem's actors affect, and are affected by, the creation and delivery of each other's offerings. Thus, the business ecosystem is composed of inter-member flows of material, energy, knowledge and money [11]. The ecosystem may emerge spontaneously due to a common interest or demand, or as a result of long-term strategic planning. The members share the common ecosystem regulation but are able to act independently, and join and leave the ecosystem freely, since there is no dependency between ecosystem members.

A service ecosystem is a socio-technical complex system that enables service providers to reach shared goals and gain added value by utilising the services of other members in the ecosystem [17,22]. Digital service ecosystem is a part of a service ecosystem, but only covers the digital part, leaving out the purely social part. Digital service ecosystem can be characterised according to [1], being an open, loosely coupled, domain-clustered, demand-driven, self-organising agents' environment, in which each species (human, economic species and digital species, i.e. computer, software and application) is proactive and responsive for its own benefit or profit. The product of a digital service ecosystem is a digital service that is entirely automated and that can be anything that can be delivered through an information infrastructure, e.g. web, mobile devices or any other forms of delivery. For example, the digital service ecosystem can provide the devices and applications as services used by a medical team, but not the whole treatment process (including doctors, nurses, etc.) is provided as a service. Similarly, as in a business ecosystem, partner networks are created inside a digital service ecosystem, but there also exist other dependencies between the ecosystem members than business dependences. The members share the service taxonomy and service descriptions that can be categorised, for example, by domain, purpose or technology. The focus is on dynamic, behaviour and conceptual interoperability and interactions between services, and between humans and services.

A software ecosystem has some common elements with digital service ecosystems, such as self-regulation, networked character and shared value [14,15]. However, the definition of a software ecosystem suggests that in general there will be some technology underpinning the ecosystem [13–15], whereas in a digital service ecosystem the members are not bound to a shared development platform or technology. Business and digital service ecosystems can be created dynamically, whereas in software ecosystems, a common platform is required to be developed first. A software ecosystem can be a part of a digital service ecosystem, but in that case the software must be provided as a service to the ecosystem. In a software ecosystem the focus is on technical, syntactic and semantic interoperability and interactions between systems and humans, and there is an increased dependency between ecosystem members.

2.2 Challenges of ecosystem-based service requirements engineering

The importance of service innovation has become the key issue due to dynamicity in customer demand, faster time-to-market, increased competition and the possibilities of co-creation in value networks. Open innovation breaks the boundaries around a company in the innovation phase; the companies can create ideas by themselves and use external ideas or co-create ideas with other companies or the actors of other communities. Due to these characteristics, open innovation is well-suitable for ecosystems. Service innovation can have two forms; outside-in and inside-out [23]. Outside-in innovation is required in cross-domain service engineering and in open data ecosystems that freely exploit the available data. Inside-out innovation focuses on opening internal data, not useful as such for its provider, for other actors' use or for sharing service ideas that an inventor is unable to develop by

himself/herself. In the outside-in process, the external knowledge and innovation components are used in service development, whereas in the inside-out process, the company allows external parties to use its knowledge and innovation components in the service development. In an ecosystem, the value is co-created in a value network, which can be formed as a result of long-term co-operation, or dynamically among members to reach the solution. Several value networks co-exist inside the ecosystem. Value networks can be formed already during the service innovation phase, when each actor has his/her own interest in the service.

As the digital service ecosystem enables members to utilise the methods and technologies that best suit their own needs, two main elements are required to be defined and provided by the ecosystem to engineer services in an ecosystem:

- *The ecosystem infrastructure* is required to make services interoperable, available, and easily consumed and thus manage all service ecosystem operations [22,24,25].
- *The knowledge repositories* are required for storage of the collaboration models, service descriptions and ontologies of service types to support interoperability validation [25–28].

The intent of the knowledge management model is to guarantee the effectiveness of the service ecosystem by maximising semantic interoperability and alignment among ecosystem members, services and technologies. The knowledge base, including business know-how, assets, architectural knowledge and tooling, is required and exploited in each service engineering phase.

In summary, four main challenges for ecosystem-based service RE can be identified:

- Service co-innovation: The open innovation between ecosystem members to identify ideas for a service must be enabled.
- Service value co-creation: The co-creation of value in the value network must be enabled by utilising the ecosystem's rules, methods and practises for service engineering.
- Enabling infrastructure: The infrastructure with the support for service collaboration and co-operation of ecosystem members must be provided.
- Utilisation of ecosystem's assets: The existing ecosystem assets must be able to be utilised.

2.3 The service requirements engineering methods for ecosystems

Several surveys and reviews of the RE frameworks, approaches and methods have been concluded recently, such as [29–33]. In [32], Service-Oriented RE and the Scenario-Based RE are defined as emerging trends in RE. Despite the research results made in the context of service-oriented models and techniques, there is still a need for new techniques and approaches for RE activities [2,4].

2.3.1 Service co-innovation

Service innovation is already taken into account in the recent research on service ecosystems. In [34], a common underlying architecture is used to connect different pieces of innovation components, and it also considers the value proposition for different participating partners. An open government data portal in [35] is also used as an open innovation platform to attract businesses and citizens to create e-services. An innovation framework introduced in [36] supports the development of new services through integrating customers, suppliers, complementors and competitors. A conceptual framework for web-service ecosystems proposed in [16] emphasises a central platform from which the companies try to extract ideas for service innovation and use these ideas to create new, or improve existing, services. The structure of the value chain affects innovation, requirements engineering performance and software success, such as described in [37]. A method introduced in [38] emphasises socio-technical aspects such as context, environment, and team management in service innovation. In [39], two types of requirement engineering methods are suggested to emerge for IT services: RE for *service consumers* and RE for *service providers*. Consumers will focus on identifying the tasks that need support, whereas providers will focus on achieving economies of scale such as offering a new service for multiple customers, or applying a specialised skill to a common problem. In several approaches, such as [29,40–43], the identification of business goals and the business processes that support those goals are used as a starting point for the RE of services. Several approaches for the requirements engineering of service users are also suggested, such as [44–46]. Both these viewpoints are required in ecosystem, since the usage goals of consumers and the business goals of service providers must be fulfilled by the services. The fact is that the current approaches and methods lack of tool support, they do not cover all the phases of RE, or they concentrate only presenting only a technique applicable in a certain RE phase [33].

2.3.2 Value co-creation

Although the significance of open innovation has been detected in the context of the ecosystem, there is not much research in the literature on how to go further from ecosystem-based service innovation to service co-creation inside an ecosystem. However, some approaches exist that deal closely with the subject. The Inter-enterprise Service

Engineering Framework [47] supports three phases of e-service development in business ecosystems: requirements analysis, service design and service implementation, assigning them to strategic, conceptual, logical and technical abstraction layers. Service requirements are identified in the strategic perspective in the form of a business model. However, the service ideas are innovated, identified and evaluated and their business potential is analysed prior to the development of the strategic perspective, but the approach does not define how this is done. In [48], an RE approach is introduced, which uses guided questionnaires to elicit the requirements coming from the current business situation, and a workshop to define the basic requirements for each Manufacturing Service Ecosystem scenario. The approach enables the relevant ecosystem actors to participate in scenario identification and requirement elicitation, but it does not define the content of RE phases and the impact of other ecosystem elements on RE. In [36], there exists a mapping of information collected in the Innovation Repository accessible to service engineering, but the approach does not describe how this information affects to service realisation.

2.3.3 Enabling infrastructure

The interoperability models and rules enable the loosely coupled services to collaborate. In [25], six interoperability levels are defined for smart environments: conceptual, behavioural, dynamic, semantic, communication and connection. In [49], four inter-related metamodels are suggested for ecosystem interoperability: domain ontology, methodology, domain reference, and knowledge management metamodels. In addition to service interoperability, pragmatic interoperability [49] is achieved between ecosystem members when their intentions, business rules and organisational policies are compatible. To detect service interoperability, the service must be specified in an understandable way in the ecosystem's service registry. Three levels of service specifications can be identified [26]. The importance of interoperability models has been recognised in the context of ecosystems; still there is a lack of application of these models in practise.

2.3.4 Utilisation of ecosystem's assets

The knowledge models are usually described using ontology models, which are guided by knowledge management. The knowledge- and ontology-based requirements engineering has a long history [50]. Even currently, an increasing amount of research has been conducted to utilise ontologies in RE [51]. Different kind of approaches have been suggested, such as for generating a requirements model based on the concepts in the service requirements modeling ontology [52] or establishing a mapping between a requirements specification and ontological elements [53]. A knowledge-based development of intelligent smart spaces is introduced in [27], which support the service innovation and co-creation, exploiting the usage scenario description, the set of use cases that define the specific viewpoint of the usage scenario, and the reusable artefacts, such as ontologies, models, patterns and rules, provided in the knowledge base. In [25,54], an approach for developing intelligent applications/services for smart spaces is introduced that exploits the ontology models, interoperability models and context models for describing self-adaptable services. The approach is multi-technology and multi-domain oriented, but it still lacks the business (ecosystem) viewpoint.

2.3.5 Summary

We can notice that there already exist several methods and approaches that enable some parts of the ecosystem-based RE for services. Several innovation methods exist for services that enable the open innovation among ecosystem members. Several models have been introduced to verify interoperability in ecosystems. Also, support is provided for knowledge-based service engineering that would enable to utilise the existing assets of the ecosystem. However, the interoperability models and knowledge base service engineering have not been present in methods for ecosystem-based service engineering. Service innovation approaches either do not take these into account. Furthermore, the innovation approaches are separated from the further phases of service development. The identified methods and approaches for ecosystem-based service development are loose; they are only concentrating their own viewpoint and not working together. Clearly there still exists a lack of methods how to take the digital service ecosystem elements into account in service engineering, especially when they have a direct effect on RE of services. The diversity of RE approaches and their use in different contexts highlights the importance of the formal definition of terms and the definition of uniform modelling methods and techniques. The identified challenges for service RE [2–7] and for service RE in ecosystem [8,9] still remains, meaning that the service RE is not mature enough as such and not especially for digital service ecosystems. There are multiple actors, viewpoints and ecosystem capabilities that affect the service RE in an ecosystem, but at this moment, there is no RE method for digital services that take these account.

3 Scenario-based service requirements engineering in a digital service ecosystem

Based on the presented state-of-the-art analysis, we have defined a service engineering model for digital service ecosystems, concentrating on the RE of services. Ser-

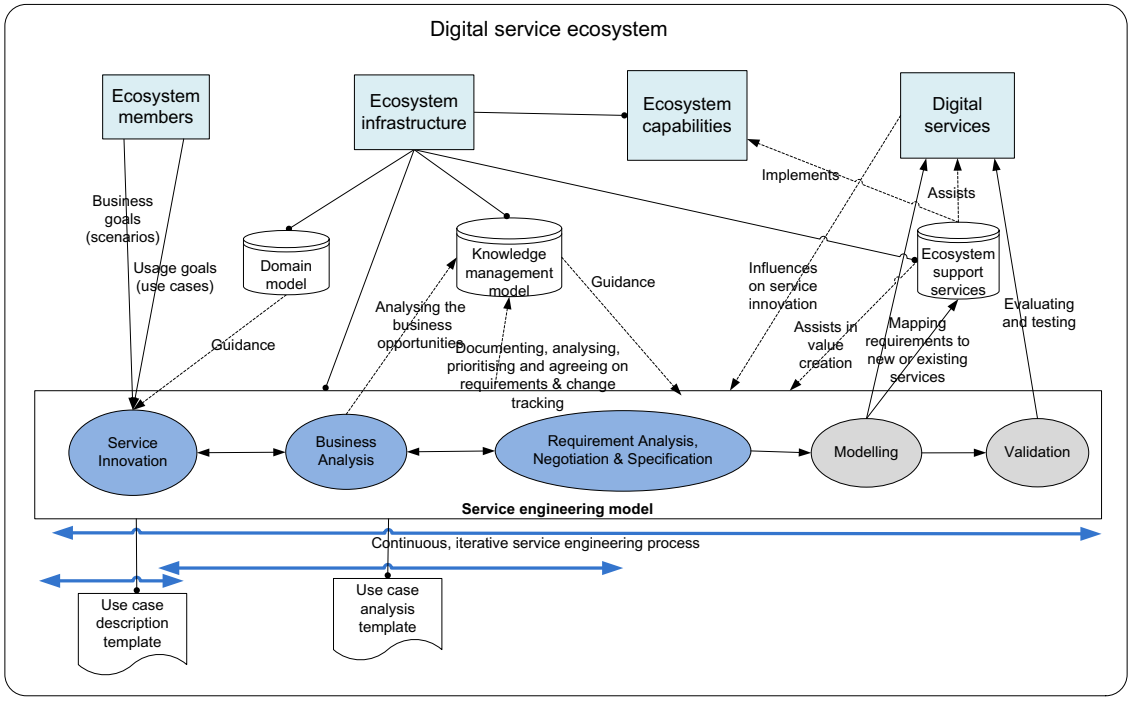

Fig. 2 The elements and phases of service RE in a digital service ecosystem

vice requirements can be classified into functional, non-functional and business requirements and constraints. Functional requirements describe the behaviour of a service that support the tasks or activities of the user. Non-functional requirements describe the qualities of the system, which can be defined as externally and internally observable properties of software systems [55]. From the non-functional requirements viewpoint, quality is regarded as constraints exhibited over the functionality of the service. Business requirements assist the service provider in achieving the business goals. Constraints are those characteristics that limit the development and use of the service.

Figure 2 describes the service RE elements and phases in a digital service ecosystem, which are introduced in more detailed in Sects. 3.1 and 3.2. The last two phases in Fig. 2, modelling and validation, are our ongoing work and are described in our upcoming paper on service architecture design and are therefore out of the scope of this paper.

3.1 Digital service ecosystem elements

3.1.1 Ecosystem members

The members of the digital service ecosystem can be defined to include service providers, service brokers, service consumers and infrastructure providers. Service consumers use the services and define the usage goals for the services, i.e. tasks that need support. They may also report on problems and failures in the service usage and provide feedback for the service validation. Service providers are independent members that provide digital services to be used by other ecosystem members or consumers. Service brokers promote the services, enable service delivery and match the demand with the best available services. Infrastructure providers provide services that implement the purpose and capabilities of the ecosystem, such as establishing, modifying, monitoring and terminating collaborations, and operations for joining and leaving collaborations. The ecosystem can also include other infrastructure provider roles, such as service market-place providers, tool providers, cloud service providers and interface providers [56]. The ecosystem provider is usually the key organisation, which establishes and maintains the ecosystem, controlling its function, members and services.

3.1.2 Ecosystem capabilities

The ecosystem capabilities describe the capability model that defines the properties of the ecosystem, and how these are implemented using the ecosystem services that the ecosystem infrastructure provides. The capabilities define the purpose of the ecosystem, its ability to perform actions and the rules of how to operate in the ecosystem. The capabilities define the governance activities and regulation processes for the ecosystem for directing, monitoring and managing the ecosystem. These include, for example, how a trusted collaboration can be established between members, what the interaction rules

are, how to join and leave the ecosystem, and how to describe and deliver services. In addition, the ecosystem capabilities define how the knowledge is managed.

3.1.3 Ecosystem infrastructure

Infrastructure implements ecosystem capabilities, supporting the utilisation of core competencies and core assets, flexible business networking, and efficient business decision-making. The infrastructure also includes the taxonomy and shared descriptions of services (categorised by domain, purpose and technology etc.). The infrastructure provides the following models and assets that assist in the RE:

- *The domain model* describes the concepts of the domain and the relationships between those concepts. The domain model can be used for configuring and adapting service artefacts for use in other domains. Thus, it supports evolution of the service ecosystem.
- *Knowledge management model* enables reuse of the existing knowledge on the business, and design practices and existing assets in the development of new service, maximising semantic interoperability and alignment among ecosystem members, services and technologies. For example, quality ontologies define the concepts, relations, rules and their instances, which comprise the relevant knowledge of a topic and assists in reaching quality requirements, and quality-driven design methods enable to achieve the quality requirements in the architecture.
- *The service engineering model* describes and guides how the services are being engineered in service ecosystem, assisting in innovating, analysing, modelling and documenting requirements. The scenario-based RE technique is chosen, because it enables to describe both of the viewpoints of RE: business and usage. The main activities are supported by the two templates: The Use Case Description template is used for service innovation and The Use Case Analysis template assists in identifying, analysing and specifying the requirements.
- *Ecosystem support services* are responsible for providing tool support for the activities of the service engineering, e.g. for using quality ontologies in all development phases (design, implementation, and testing). In addition, support services assist its members in value creation, for example, by contract making, finding partners, services, and/or markets, and analysing business models [56]. The infrastructure should also provide collaboration support services for ecosystem members, e.g. for communication and document sharing. The ecosystem should be aware of the quality of the services in the ecosystem's service registry; therefore, the ecosystem should also include support services for run-time quality monitoring and analysis of services [25,30]. Furthermore, the service registry should provide feedback mechanisms for users to provide feedback about the service, thus supporting requirements change and evolution.

3.1.4 Digital services

In a digital service ecosystem, digital services are provided by independent ecosystem members, where they provide additional value for both service consumers and other service providers. Service providers do not necessarily provide a complete service for consumers but can just provide part of a composite service [57]. Service level agreements (SLAs) are negotiated between the atomic service providers and the composite service provider, describing the agreed-upon terms with respect to quality of service and other related concerns. The results of the RE process, i.e. service requirements, either result in new digital services, or they are mapped to existing digital services. The requirements can also be identified as new ecosystem support services, or they can cause changes to the existing ones.

3.2 Service RE phases

Service RE in a digital service ecosystem is an iterative process consisting of three phases. These are described in the following sub-sections. Before the RE can begin, the following activities need to be performed inside the digital service ecosystem:

- Identifying the value networks: The actors that have their interest in the service are identified and their contributions in the value network are defined.
- Identifying roles in RE: The roles and responsibilities in service co-innovation and co-creation are defined for each member considering all activities in RE.

3.2.1 Service innovation

The service innovation phase starts the service RE in digital service ecosystem. The main purpose is to identify the ideas for new services, scope and analyse them and transform them into service requirements. The innovation can be divided into two sub-phases: requirements elicitation and the requirements identification of services.

Requirements elicitation

Purpose: Requirements elicitation is a practice of obtaining the requirements from all stakeholders. The requirement elicitation method defines what, how and from whom the requirements should be elicited, and guides the elicitation process.

Activities: The elicitation process can be divided into the following steps:

(i) Identifying responsibilities: The members responsible for the activities of the elicitation process, such as coordination, requirements collecting and management, are identified.
(ii) Identifying ecosystem assets: The domain model(s) of the service ecosystem assist(s) in understanding the relevant domain concepts that affect the service RE. The knowledge management model provides knowledge, know-how and assets, whereas the capability model describes the ecosystem support services, tools and guidance for RE.
(iii) Identifying requirements sources: service provider identifies requirements from the business goals: service consumers provide ideas, requirements or feedback for services, service broker helps in service delivery and mediates between service providers and consumers; and ecosystem infrastructure defines domain ontologies, knowledge management models, etc.
(iv) Analysing stakeholders: This analysis extends the stakeholders from business, user, technical and management points of view. The analyst shall have a clear vision and understanding of what kinds of stakeholders are to be covered and what are to be scoped out.
(v) Introducing the approach and tools: The approach used for a service RE is a scenario and use case description-based elicitation approach. Business scenarios are described by the service provider that wants to create value and achieve economic returns with the help of the service. The scenarios are refined into use cases that describe the user's point of view. Several UML-compliant tools enable the definition of use case diagrams.
(vi) Eliciting the requirements: The Use Case Description Template is a Microsoft Word document template that assists in identifying and documenting the motivation of the use case, and inspecting and documenting the use case from the following viewpoints: contextual, functional, non-functional, business and constraints and threats.

Practice: The domain model(s) describes the relevant domain concepts that should be considered in the RE. The knowledge management model provides ontologies, methods, tools, templates and guidelines for requirement elicitation. The Use Case Description Template ensures that all the use cases are described adequately, consistently and with the same accuracy. The use of the template is to make it easier to transform from use cases to identifying services and their relationships. Functional properties can be identified from the use case flow, which is a detailed description of the user actions and system responses during the execution of the use case. All the elements of the use case, i.e. actors, the use case function, actor-use case relations and the use case environment, should also be considered from the non-functional viewpoint, which describes the quality properties of the use case. For example, reliability properties describe the issues affecting the failure free operation of the use case (how the fault prevention, tolerance, removal and recovery from failures are considered in the use case), and availability properties describe the issues ensuring that the use case function is ready for use when required. The Use Case Description Template includes the classification of quality properties and their detailed description to assists users to consider each quality property.

Outcome: The outcome of the phase consists of the following for each use case: *General information*: the identification, introduction and rationale of the use case,*Contextual settings*: A description of the context of the use case, including actors, a vision of the infrastructure, the physical resources and required artefacts, *Functional and non-functional description*: the description of the main functions and the quality properties of the use case, *Business properties*: a description of business properties or the use case, such as customer segments, value proposition, customer relationships, etc., *Constraints*: a description of the constraints associated with the elements of the use case, and *Threats and exceptions*: a description of the threats for the use case, responds to those threats, and description of anticipated exceptions and error conditions.

Service requirements identification

Purpose: The purpose of the requirement identification phase is to identify, classify, merge and prioritise service requirements using the use cases defined in the previous phase as input.

Activities: Each member identifies one or more functionalities for the use case, and identifies and maps the functional requirements to each functionality.

Practice: The Use Case Analysis template assists ecosystem members in this activity, using as input the information gathered in the previous phase. The template enables to analyse the use cases from business and user points of view and to identify and describe functional and non-functional requirements and constraints. The template also assists and guides in making an initial mapping between the identified requirements and the existing digital services and potential new required ones. Figure 3 describes the mapping with the Microsoft Word smart-art tree diagram provided in the template. The owner of the use case plays a key role in mapping the identified requirements to the support services provided by the use case partners involved in this particular use case. Each functional requirement results in one or more 'support services' that implement the requirement. The support

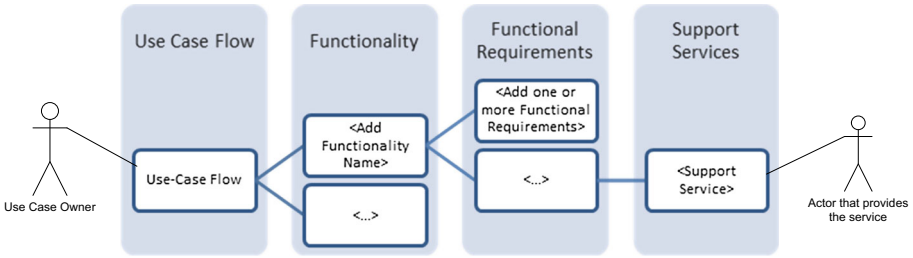

Fig. 3 Service requirements identification and mapping requirements to services

service is a new or existing that provides the functionality and quality that is required to implement the use case. If the requirements cannot be mapped to already existing services, totally new services are identified here, which are responsible for implementing the requirements. When candidate services have been identified, non-functional requirements can be mapped to them. This especially concerns the execution qualities, such as performance, availability, reliability and security, but evolution qualities (such as reusability, modifiability and maintainability) are focused on later on in the service modelling phase, where the architectural knowledge guides the design work.

Outcome: The outcome of this phase includes a detailed description of supporting services identified in use case analysis, including functional requirements, non-functional requirements, data resources, constraints and their mappings to the identified support services.

3.2.2 Business analysis

Purpose: The goal of this phase is to identify which use cases have the most business potential in the ecosystem, i.e. the business analysis helps to define what requirements to implement, and how.

Activities: The identified use cases are collected together in the ecosystem and the members responsible perform the business analysis. Therefore, each use case is analysed by several analysts according to the following criteria (defined in the knowledge management model of the ecosystem) [27]:

1. Maximum business impact (rated: 1–10)
   - Added value: The usefulness to customer is higher than adoption effort and costs;
   - Partners interest: The business opportunities for ecosystem members;
   - Market penetration: The time period the solution can be brought to the market.
2. Fast and low-risk realisation (rated: 1–10)
   - Availability of technology: compatible with and builds on popular legacy technologies and assets;
   - Implementation complexity: nature and amount of R&D effort is known and feasible.

The given numbers can be weighted to be appropriate for the case. After the business potential analysis, the most relevant use cases from the business viewpoint are identified, and the initial mapping of the identified requirements of these use cases to the responsible services (according to Fig. 3) are verified. At this point, the actors that provide the required support services (see Fig. 3) are identified. Some of the requirements can be implemented by already existing services, when contracts are being made with other service providers, whereas some of the requirements result in identifying new services.

Practice: The knowledge base of the digital service ecosystem contains business knowledge, architectural knowledge, assets and supporting knowledge management tools [27]. Business knowledge also provides information about the earlier ideas that have been evaluated but not realised. The reasons behind the earlier decision are valuable in making feasibility checks while further defining the usage scenario and use cases in hand. The knowledge base also includes documentation about existing assets that resolve their availability, suitability and quality for the service development in hand. The architectural knowledge is used to estimate what artefacts can be used as such, or modified, what is the quality of the artefacts and whether the quality is satisfactory.

Outcome: The outcome of this phase is a set of use cases, the related requirements and the analysis results on business impact and risks related to the use cases.

3.2.3 Requirements analysis, negotiation and specification

The last phase of the service RE includes three sub-phases that are highly interrelated and iterative by nature. The main purpose of the phase is to provide a complete requirement specification of the needed services that is used as input in service architecture modelling.

Service requirements analysis

Purpose: The goal of requirement analysis is to determine the consistency and completeness of requirements, and also

the priority of requirements. The purpose is to analyse the capabilities and constraints of the existing services, different potential technologies for service creation and the contribution of the service to the different business cases defined in the earlier phase.

Activities: The starting point of this phase is the existing service architecture and/or description of services, or a sketch of new service architecture. The candidate services for the use case identified during service requirement identification and business analysis are listed and checked with partners. The similar services are merged and those with no business potential are rejected. The taxonomy and shared descriptions of services provided by ecosystem infrastructure enables to categorise services by technology, domain and purpose. The classification according to taxonomy assists in getting better understanding about requirements. For each service, the requirements are analysed, and combined if necessary. As a result, two different services might be required for different users. Thus, a variability analysis is made from four viewpoints: the service provider, the service user, the usage context and the implementation technology. The trade-off analysis is performed for conflicting requirements according to their importance and the results of the business analysis. Also, quality requirements are prioritised based on their importance to stakeholders and as a result of the trade-off analysis.

Practice: The knowledge base includes rules for trade-off analysis, e.g. the rules for ranking quality attributes. Each quality attribute is a representation of a single aspect or construct of a quality. The Use Case Analysis template enables the detection of the requirements mapped to each service, and the importance for each requirement.

Outcome: The updated architecture/vision of the services, and the analysed and prioritised quality requirements for each service.

Service requirements negotiation

Purpose: The purpose of requirement negotiation is to communicate the service requirements to the business and technical stakeholders involved in service development and agree the service requirement specification with the ecosystem members. This means active collaboration among the members of the digital service ecosystem by exploiting the ecosystem infrastructure.

Activities: The starting point of this phase is the analysed service requirements description that gives the first proposal of the balanced service requirements. E-mail and collaboration support services (e.g. document share points, co-design tools, video conferences and telecommunications) can be used for communication and negotiation between ecosystem members. Voting is rarely needed for design decisions, but it is also possible to be organised by using the digital ecosystem support services. Typically, several negotiations are carried out with different focal points: e.g. big picture, domain variations and business and implementation constraints. That is why several rounds of the analysis-negotiation-specification loop are required.

Practice: The domain model assists in getting common understanding. The knowledge model assists in sharing evolving knowledge and specifications. The knowledge base includes guidelines on how the service negotiation is to be carried out and how the design decisions are to be documented and recorded for future needs. Due to service evolution, special attention is to be paid to documenting the agreed design decisions, the proposals that have been discarded and the reasons behind the decisions. The ecosystem support services (automated or practical guides) assisting in negotiation among the ecosystem members. The Use Case Analysis template can be used to document the design decisions but tools that provide automation support are preferred.

Outcome: Negotiated service requirements and a list of agreed and rejected service requirements and the design rationale behind the decisions.

Service requirements specification

Purpose: The purpose of this sub-phase is to describe the service requirements specification by using textual and graphical notations that make requirements specifications complete, understandable and useful for all ecosystem members. The requirements specification is a complete description of the behaviour of the services to be developed.

Activities: Several rotations are required to illustrate domain requirements, business requirements and technical requirements for services. The first round ends with an initial service taxonomy that defines what kind of digital services are needed and clusters them to the groups of services based on their usage or/and technical relations. The last round ends with service specifications that provide a big picture as a set of master use cases that describe the behavioural service architecture. A master use case is made by integrating the related use cases defined by different business actors (see Fig. 3, use case owner). The master use cases give a mutual understanding of the service architecture and how it is used for realising the different use cases by diverse actors. Thus, the service requirements specification includes the activities that transform informal service specifications into semi-formal descriptions to be used as a starting point in the service architecture modelling phase.

Practice: The Use Case Analysis Template assists in describing the requirements in a consistent, accurate way. The

Table 1 Preliminary set of digital services in the ICARE ecosystem

Acronym	Description
PRM	*The Processing Resource Manager* is in charge of content ingest and cloud resources management (e.g. load balancing)
BWM	*The BandWidth Manager* regulates the networks according to traffic overload and user requests
SCM	*The Security Content Manager* controls the networks to get a good QoS and guarantees that content is delivered at the right place at the right time
CDM	*The Content Database Manager* can be used for publishing and retrieving content. It knows content properties in the cloud infrastructure and can retrieve them for play-out delivery
MAM	*The Media Asset Manager* (MAM) and its compounds (i) handle the descriptive metadata that are delivered along with the Media assets, (ii) manage the work orders for traffic managers and video engineers, and (iii) manage the deep archive, the transcoding and the delivery

description of the service taxonomy and services is guided by the knowledge management model.

Outcome: A service taxonomy and a set of master use cases that describe how the digital services—to be developed and existing ones—interact and cooperate with each other in order to provide the required end-to-end digital services. The updated architecture/vision of the digital services, and the analysed, prioritised and agreed requirements for each service.

Since the service engineering model is iterative (described as double-headed arrow in Fig. 2), it takes into account the requirements evolution caused by new requirements, feedback achieved from the use of the service, and change requests in current requirements. The identified new or changed requirements proceed from the service innovation to the service identification, where the requirements are mapped to the existing digital services and potential new ones. The requirements go normally through the business analysis and requirements analysis, negotiation and specification. It is the responsibility of the modelling phase (out of scope of this paper) to decide how to implement the new and changed requirements.

4 Example of using the digital service RE method

So far, the RE method has been in use in two different cases. This section describes the first usage of the service RE method, when it was applied to specifying the digital services and related support services for an interactive multi-screen TV services ecosystem in the Innovative Cloud Architecture for Real Entertainment (ITEA2-ICARE) project.[1] This digital service ecosystem includes 25 service ecosystem members from Europe providing and using digital cloud-based services related to the operation of end-to-end interactive multi-screen TV services. The ecosystem member roles include, for example, multi-modal content service providers, communication infrastructure service providers, cloud platform service providers and consumer cloud service providers. The ecosystem services include content processing, multi-channel user interaction and content access management services, which are all needed as part of operation in the final end-to-end TV service provided to end users.

The goal for applying the RE method was to collect and analyse requirements from the ecosystem members towards a shared service-oriented platform enabling the provisioning, integration and use of services amongst the members of the ecosystem. An additional aim was to support ecosystem members in specifying their digital service offerings and needs via describing use cases with business model analysis within the digital service ecosystem from the viewpoint of an individual ecosystem member. In the early phase of the ICARE project preparation, the domain model of cloud-based multi-media services was sketched and analysed by the collaboration partners. As a result, a preliminary set of digital services for the ICARE ecosystem was proposed (Table 1). These service descriptions are abstract and intended to give a mutual understanding the purpose and ability of the ICARE ecosystem. These services are covered by the existing services and new services provided by the ecosystem members.

4.1 Service innovation

4.1.1 Requirement elicitation

The Use Case Description Template in the form of Microsoft Word-file was used for collecting input from the ICARE ecosystem members. Thus, the usage of the template did not require special tools. In addition, guidance was provided in the form of instructions for using the templates. The instruction included the definition of the main elements, description of the purpose and goal of the Use Case Description and common instructions for fulfilling the templates. Both business and technical experts were asked to be involved in defining use cases. However, only one contact point was defined by

[1] https://itea3.org/project/icare.html.

Table 2 General information of the second-screen voting use case

Summary	This use cases defines how mobile phone-based services can be linked with live TV broadcast. An end user has a mobile application related to a TV programme. The user gets some insight into the upcoming shows with the application. Also during the live TV programme interactive voting services will be provided to an end user. This interactive voting service contains advertisements that can be personalised
Rationale	People are watching less and less live TV programmes. This can be problematic from the advertiser and broadcaster point of view. Providing new ways for people to be committed to live TV programmes can make advertising more efficient. Also the user commitment to TV programmes can be increased. The user will spend more time with using TV and the additional services
Description	1. Alice installs the Talent application from the programme's web page 2. Alice uses the application to get some insight into the upcoming shows before the live show 3. Alice is watching the live Talent programme 4. After the programme, a competitor notification is sent to all watchers who have installed the application 5. The notification includes a voting form in which the user can give 1–5 star rating to the current competitor. The notification also contains ads which can be personalised 6. Alice saves the discount coupon for her favourite store shop, which is included in the notification 7. After saving the coupon, Alice rates the programme and presses the submit button 8. On the live TV programme, the average user ratings are shown

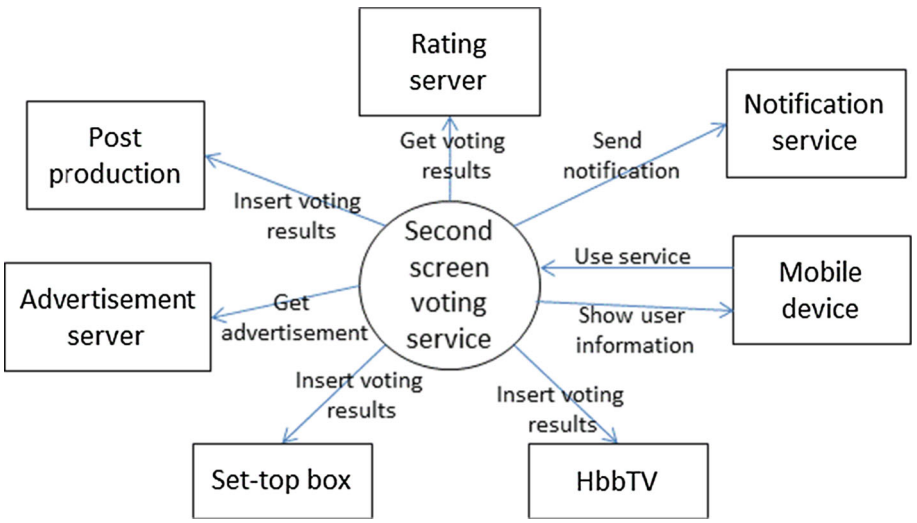

Fig. 4 Second-screen voting use case context diagram

each partner, and it was not visible to other ecosystem members which kind of expertise was used to define the use cases.

All in all 41 use cases were defined. Some of them were variations on a similar theme, the actual number of different use cases being about 30. Filling a use case took about a week by each partner, but it took 3 months to collect all of these use case descriptions because of the participants' work schedules and the need to iterate the use cases. The activities of the ecosystem members were divided according to work package descriptions of the project; thus, one partner was responsible for collecting and coordinating the use cases. To illustrate the outcome of the Service Innovation phase, the definition of the 'second-screen voting' service defined by an ecosystem member is used here as a practical example.

General information: ICARE UC No. 4 second-screen voting

The general information of the use case is represented in Table 2

Contextual settings

A context diagram Figure 4 describes the context diagram of the second-screen voting use case.

Actors: The identified use case actors and their responsibilities included the following: End user—uses an interactive

TV service via a mobile device application; Advertiser—provides advertisements to the broadcaster to be delivered to the end users; and Broadcaster—provides additional information to show based on the user interaction and provides an interactive service to end user.

Resources: The number of needs of physical resources and locations is more or less an implementation-specific issue. However, some parts of the system, such as software components for a set-top box and HbbTV will be in the home environment running on previously mentioned devices. The notification service and the rating service can run in the same server. Advertisements are most likely provided by third-party actors from their own servers. In addition, a post-production server must be in its own physical environment.

Frequency of use: The frequency of the usage depends on the content and users. Average usage could be 5–8 voting (notifications) during a one-hour programme. If the user base is large, there could be 10–100 thousand simultaneous notification and voting results to be handled.

Description of service properties

The Use Case Description template assisted in describing the use case, and the service innovation phase in the case of the 'second-screen voting' service resulted in the service description represented in Table 3.

4.1.2 Service requirement identification

After all use cases were identified and described, each partner identified the functional requirements of the use case, starting from normal flows using Microsoft Word's smart-art tree-based diagram provided in the Use Case Analysis template. The running ID of each identified requirement and potential support services are also shown in the diagram. In the example diagram in Fig. 5, the use case owner identified two new services to be implemented and also potential for utilising partner-provided or existing technologies to complete the use case. The requirements shown in the diagram were then specified in more detail. The use case number and use case-specific requirement ID was used as the global ID for each requirement. The detailed description is provided in Table 4. The importance (imp) of the requirement ranges from 1 to 5, (1 = not relevant). The template also provided the possibility to define non-functional requirements and constraints, and associate them with functional requirements. For example, two availability requirements were associated with the functional requirements 4.1 and 4.2 (see Table 4). The category (cat) describes the type of the requirement (F = functional, NF = non-functional, C = constraint). The table was scalable and could be complemented in more detailed afterwards; the 'Details' column allowed to insert more information.

Several data resources were also identified: TV programme info, Notification content, Voting results and User profiles. These are specified in the data resource definition template with characteristics of resource name, description, related requirement(s), standards, quantity and size, privacy and details.

4.2 Business analysis

Table 5 gives an example how the business impact and risk analysis results could be collected and compared in order to make decisions on how to proceed and with what use cases. Both the business impact and risk analysis were rated in a range of 1–10. As can be seen, the development should be started from UC No. 1 if maximum impact with low risk (i.e. availability of technology and implementation complexity) is preferred. Also, it can be seen that UC No. 5 is not realistic and should be rejected.

According to Fig. 5, the identified potential for utilising partner-provided services to complete the use case enabled the finding of co-operation partners with a similar focus.

4.3 Requirements analysis, negotiation and specification

4.3.1 Requirement analysis

The services and service providers identified during service requirement identification and business analysis were collected into a list preserving the link to the original use case. The list of candidate services was checked with partners, and similar services with different naming merged and those with no business potential rejected. At this stage, the service candidates were quite heterogeneous both in scale and conceptual level. A service candidate comprised functionalities from algorithms and functions to sub-systems consisting of several servers. In order to better grasp the overall requirements for the ecosystem, the services were classified into service taxonomy based on their technology position. The services were classified into two dimensions: cloud services vs home network services and infrastructure vs end-user services. A partner interested in providing a service candidate could then check the requirements set to it and similar services tracing their links to the use case descriptions. Based on analysis of the business models presented in the use case descriptions, the shared requirements were also identified (see Table 6).

Several of these requirements could be assigned to individual use cases or service candidates. For example, the cross-distribution platform interoperability could be linked to interactive TV directly supporting different distribution platforms, and also to multi-screen interaction that can cope with emerging technologies helping the user with access to different mobile devices, advanced interaction technologies and multiple screens at home.

Table 3 Description of the service properties

Functional properties	
Preconditions and assumptions	End user must have installed the mobile application on his/her mobile phone
Trigger	Live programme starts and there is programme that requires voting
Normal flow	Users receive a voting notification. The user notices the ad in the notifications and clicks/saves it. The users vote. The end results are visible in the live programme.
Post-conditions	System contains the voting results
Non-functional properties	
Reliability	The user must have a working data connection on his/her mobile device in order to vote
Availability	All services must be up and running and linked. The live programme must be broadcasted
Performance	Sending several parallel notifications and processing the voting results are the bottle necks of the system. This means that there must be scalable hardware resources for system-critical services
Security	The application on the mobile device must be registered with a notification service. Additional information about the user or device (IMEI) can be added on the registration request
Interoperability	Services provide HTTP APIs for IOP usage
Adaptability	The notification service provides different kinds of notification types for different usage. Showing the result can be done in many ways
Variability	The content of the pushed notification can be varied based on the situation
Scalability	The notification and voting services can be implemented as cloud services
Personalisation	The mobile device needs a personal account (e.g. Google account) which is needed for registering with the service
Business properties	
Customer segment	Common people who watch TV. Probably younger ones
Value proposition	Users get some dedicated information about the programme which they cannot get elsewhere. Users can also vote for free. They can get some discount coupons
Channels	Application delivery will be done via mobile marketplaces. Marketing and promoting of the service is done during the programme and on the websites of the programme
Customer relationship	Users are committed to watching the programme interactively and have the possibility to have influence. Also, using discount coupons as an advertisement will keep up the user's interest
Revenue streams	Ad-based revenue is the main source
Key resources	Mobile device, notification and voting services
Key activities	Voting and advertisements receiving
Key partnerships	Broadcasters with advertisers
Cost structure	Almost everything can be automated. Only selling and linking the ads needs some interaction
Constraints, threats and exceptions	
Location	Mobile device (application) needs working data connection
Misuse cases	Someone might want to distort the voting results by accessing the voting server directly. However, the voting server can monitor the clients that are accessing it and prevent phony connections from non-application sources
Exceptions	Problems in the data connection can cause distortion of voting results. The voting server can conduct based on the registered clients and number of voters if there is no reason to show results. This means that no voting results are shown

Table 3 continued

Other relevant information	This application-based solution is a direct competitor for SMS voting. From the user point of view, this is a preferable solution because the voting is free and much more convenient. From a broadcast company point of view, the SMS are good source of revenue. This means that, in the application-based solution revenue must come from advertisements. It is also good to bear in mind that both solutions could be used together. SMS could be used by occasional watchers and the application-based solution is targeted at the fans of the programme

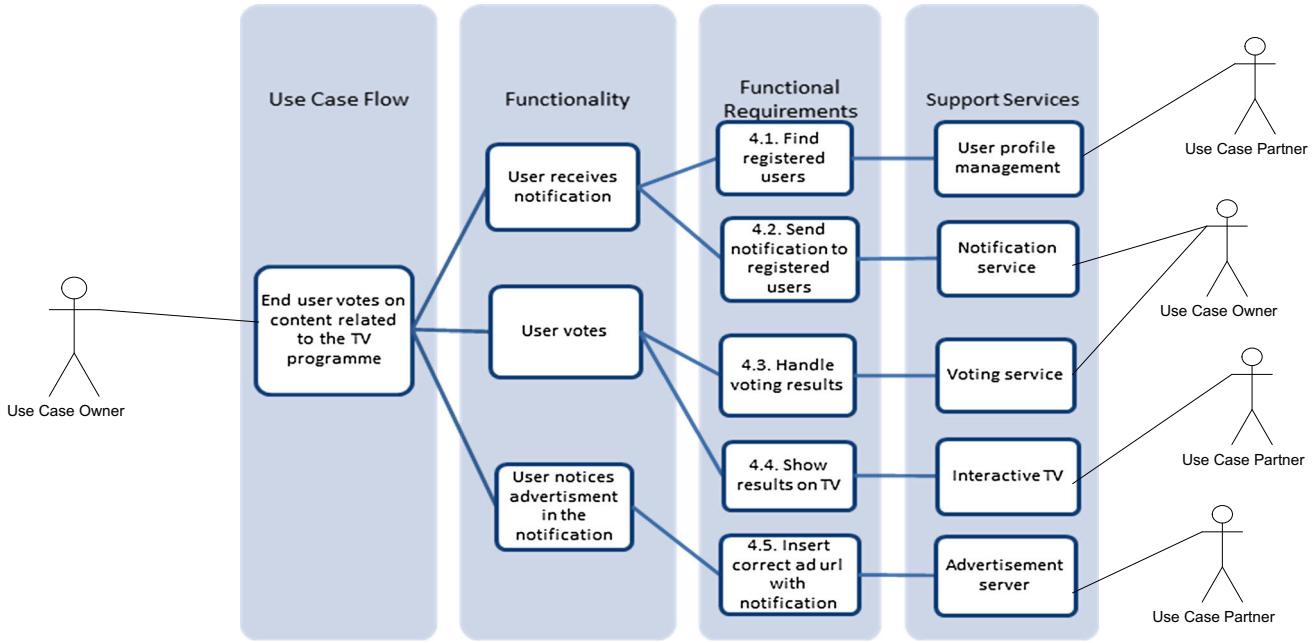

Fig. 5 Functional requirements analysis for UC no. 4 second-screen voting use case

Table 4 An example of identified use case related requirements of the Second-screen voting service

ID/req.	Imp	Description	Details	Cat
4.1	4	Find registered users to watch the programme	Input: programme name	F
			Output: List of users	
4.2	4	Send a voting notification to registered users	Input : List of users	F
4.1, 4.2	4	Users must be registered	Rationale: If users are not registered, the notification cannot be sent	NF
			Classification: Availability	
4.2	5	Advertisement have to be mapped with notifications	Rationale: Advertisement has to be mapped with the right notifications	NF
			Classification: Availability	

4.3.2 Requirements negotiation

The requirements were negotiated in smaller groups of four to five partners with a similar focus and potentially compatible business models. An overall master scenario (i.e. a 'big picture') between those partners was drafted and iterated in workshops and details refined through e-mail discussions. The master scenario draft (Fig. 6) shows the technical deployment of services and partner technologies and candidate services. This work resulted in changes in the ideas

Table 5 A fragment of the business analysis results

Scenario	Business impact	Availability of technology	Implementation complexity	Market penetration	Weighted sum
ICARE UC No. 1	6	2015	3	6	17
ICARE UC No. 2	7	2016	5	6	15
ICARE UC No. 4	6	2014	6	6	15
ICARE UC No. 5	6	2016	8	4	9

Table 6 An example of common business requirements derived from business model(s) and use case deliverables

Requirements	Description
Community building-related services (wiki/social /support)	These services will allow customers of the platform to share experience, receive advertisements about new features of services and content, ask for support as well as to facilitate the management of the platform. Support means for the content creation community building may needed as well
A cross-cloud/service infrastructure interoperability	The ability of the platform to utilise the services (computing, storage, support, etc.) from various cloud providers and service infrastructures
A cross-distribution platform's interoperability	Targeting different distribution platforms (traditional broadcasting, HbbTv, VOD, etc.)
Multiply the SLA options available for content /platform services consumers	SLA options can support different models of content distribution depending on customer profile and content type. Revenue streams related to content distribution can be based on various fee models (usage fee, subscription fee, etc.)

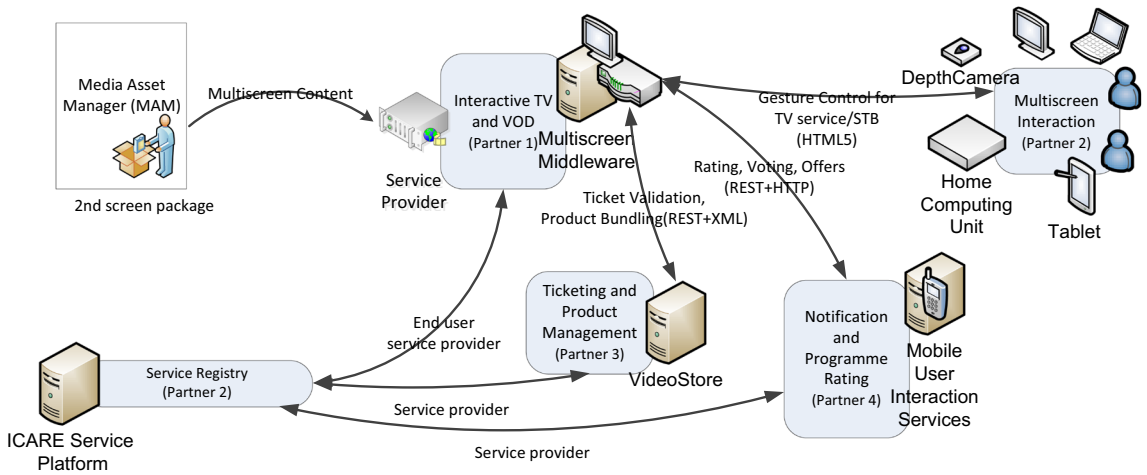

Fig. 6 The big picture describing how partners' services and technologies are related

presented by each partner in the original use cases because of better understanding of the available ecosystem services provided by the other partners. For example, there was no need for sending notification individually to the watcher device. Instead a 'red button' indication could be inserted into an interactive TV broadcast stream that a user watching the programme could select either directly on TV remote or based on advanced gesture detection. The work was documented as more or less informal architectural drawings and textual documents shared using e-mail and the master scenario was iterated until each partner was satisfied with their role in the master scenario and the support to be utilised from other partners.

4.3.3 Requirements specification

The scope of the master use case was clarified according to the master scenario and details were defined using the use

Table 7 An example of the identified functional requirements of services

Service name		Description	Service provider	Technology
Notification		Notification to the end user of events	Neusoft	REST
Req. ID	Importance	Description	Details	Category
5	5	Interactive TV Service or Multi-screen Interaction sends to the end-user device a notification of the rating	Input: User, notification content (rating link in this case)	F
			Output: Notification to the user device	
6	5	Interactive TV Service sends a notification of the reward to the end-user device	Input: User, notification content (Movie rental ticket in this case)	F
			Output: Notification to user device	

case description template with the exception that now each service provider focused only on the parts (business models and functionalities) relating to their own services. The master use case focused on the services providing interactive TV content related to the use of secondary devices. The rationale was that while use of secondary devices in conjunction with traditional broadcast TV consumption is becoming more popular, there are no methods available for personalised second-screen content that is introduced in line with the traditional content. The master use case was coordinated mainly by two active services—Interactive TV and multi-screen interaction—and consisted of several supporting services integrated with the service registry. The service dependency matrix was added to the template to aid the requirements specification. The matrix included provider services that respond to the services that implement the requirements and the dependent services that set the requirements to the provider services. The related requirements were referred with the ID number of requirements (set by dependent services). The requirements were then grouped for each service and refined as presented in Table 7. The information represented in 'details' section in Table 7 depends on the category of requirement; for functional requirements, the inputs and outputs are defined. Each partner then continued the detailed service design from there on.

4.4 Case summary

All in all, valid results were achieved in the ICARE project using the RE method. Altogether nearly 275 requirements were identified, including functional, non-functional and business requirements, and constraints. Especially in the case of the functional requirements, all the phases of the method could be performed according to guidance. The non-functional requirements, however, were seen more problematic. The definition of non-functional requirements was easy for those who had experience dealing with them, but it was clear that strong understanding about the quality issues were required. Although the templates included the description of quality attributes, that was not enough to support their use. Furthermore, the non-functional issues could be inspected from two different viewpoints, which also caused confusion, service provider's viewpoint vs. service consumer viewpoint. Different non-functional requirements were identified from these viewpoints, and the templates did not specify how to handle them. It is clear that in the case of the non-functional requirements, more support is required from the ecosystem.

5 Lessons learnt

5.1 Experiences of the usage of the method

The RE method has been applied in two different cases: in the Innovative Cloud Architecture for Real Entertainment (ITEA2-ICARE) project[2] and in the Connecting Digital Cities (EU-EIT-CDC) project.[3] To validate the usage of the service RE method in practice, we performed a feedback collection among the partners that were involved in the requirement engineering in the ICARE and CDC projects. The purpose was to receive user experiences and opinions about the method and to find out its advantages, shortcomings and development targets. The feedback collection was implemented using a web-based questionnaire that was accessible through a web page to the project partners that filled the templates. The questionnaire was implemented in April 2014 and November 2014. Altogether, we received 15 completed questionnaires. The next sub-sections describe how the users experienced the RE method.

5.1.1 The characteristics of the respondents

A total of 67 % of the people that completed the questionnaire were R&D personnel, and the rest were equally divided

[2] https://itea3.org/project/icare.html.

[3] EIT ICT Labs project No. 14465.

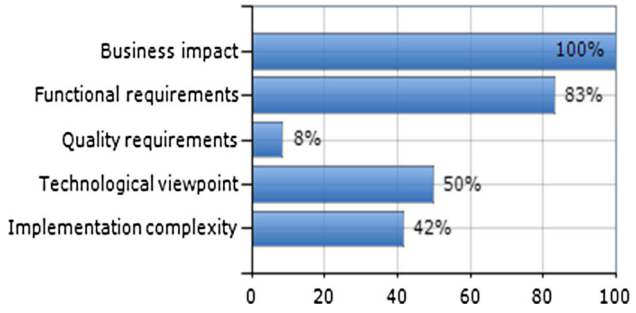

Fig. 7 The analysed viewpoints of the use cases

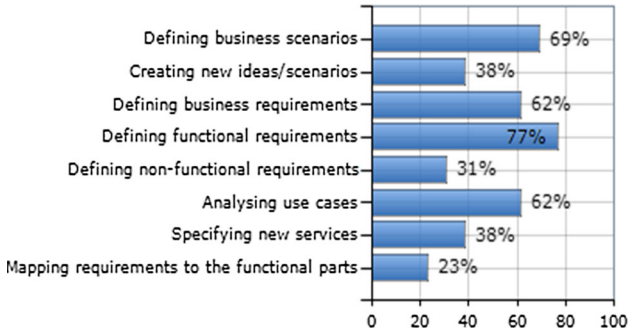

Fig. 8 The phases in which the templates were useful

between business developers and project managers. There were project managers, work package leaders, task leaders, coordinators and developers among the respondents. Five of the respondents felt that their company was a part of a service ecosystem, acting as a service provider, a technology provider, or as a GUI and platform provider. A total of 40 % of the respondents confirmed that their company utilises third-party services/technologies in software development. The target of the requirements engineering was clear enough in the project for almost all of the respondents. According to one respondent, the purpose was not explained clearly enough. One respondent felt that more information was required for focusing the scope of the project.

5.1.2 Use case definition and analysis

The definition of the business scenario was considered easy among 80 % of the respondents. The use cases were defined and analysed by architects, business managers and technical experts. The technical experts had the clear majority. Only in one case was the customer of a company involved in requirements identification (through product managers). In one case also the marketing personnel participated in use case definition. The definition of use cases from the scenario was considered easy or very easy among 93 % of the respondents. The use cases were analysed mostly from the viewpoints of business impact and functional requirements. Some also considered the quality requirements, technological viewpoint and implementation complexity viewpoints. The results are illustrated in Fig. 7. 73 % of the respondents considered the requirements easy to define from the use cases.

5.1.3 Service identification

The service identification was performed mostly by technical experts with the assistance of architects and business managers. In one case, the marketing personnel participated in the service identification phase. 73 % of the respondents considered it easy to define the services that are needed for the defined use cases. A total of 33 % of the respondents exploited existing services in defining a new service. One respondent revealed that they do not exploit existing services because they try to be innovative. The analysis and prioritisation of service requirements was seen as easy by 66 % of the respondents. The respondent companies assumed mostly cloud or platform architectures, when analysing the use cases and identifying services. The mapping of functional requirements to the architecture was considered easy for most of the respondents. Some respondents considered it difficult since the architecture was lacking high-level building blocks. The non-functional requirements were difficult to define and map to the architectural elements due to the following reasons: The vision of architecture was too complex with too many small components (lacking high-level blocks), the target and output of non-functional requirements were not clear enough, and no tool exists to support this. Almost all of the participants were able to exploit the use cases in business, at least to some extent.

5.1.4 Assessments of the RE templates

According to the respondents, the templates assisted in several phases (see Fig. 8). Also, some shortcomings of the templates were identified:

- The templates were considered to be too complex and time consuming for a small company that is trying to be agile and lean.
- The templates were too product-oriented and not technology oriented.
- The targets and outputs of the non-functional requirements were not clear.
- The title ‘Data resource’ requires a more detailed description.
- The actor description may be unclear for some people

The completion of the templates by the respondents took from a few hours (15 %), one day (62 %) to several days (23 %). For most of the respondents it was easy to apply

the templates in their working practice. It was agreed that for a large European project, documentation is a prerequisite, and the templates were good for that purpose. However, in SME companies, less formalisation and paperwork is done. For a smaller company and for internal usage, the templates are too heavy. Direct communication is preferred; a few overview slides with use case diagrams, and an ROI Excel sheet. One respondent felt that his/her work is technology oriented, but the templates are more business oriented; therefore they did not fit to the working practice. The templates included enough guidance according to most of the respondents. In one case, more guidance for the identification of new services was required. Also, working examples were required in each case that could help shape the structure and content of the proposed implementations. A more specific description of the target and output for non-functional requirements was identified as a development target. Also, more detailed explanations for the data resource titles is required. A total of 80 % of the respondents would recommend the usage of the templates to their colleagues and business partners.

5.2 Application of the method

The application of the method in ICARE and CDC projects are encapsulated in Table 8.

5.3 Summary

In the ecosystem, the co-operation between members is highly important and requires negotiation and compromise. Each member should find its own role in the ecosystem and also gain benefits in operating the ecosystem. The role of the key organisation becomes important as coordinating the business analysis, the requirement analysis, negotiation and specification between members. It is important that one member takes the role of coordinator; otherwise, the RE could result in distinct requirements and services that are not useful from the whole ecosystem's point of view. The service RE is more demanding within an ecosystem, since the RE must be coordinated at the ecosystem level and requires mutual understanding, several iterations and tight co-operation between ecosystem members.

The service RE method was seen as valuable and useful in the beginning of the service engineering process when starting the long-term development of new service architecture for digital ecosystem-based services. The service RE method was especially useful for describing, documenting and communicating the capabilities of the digital services and the service architecture they require. The method was also seen as useful in the analysis phase, where the different stakeholders work together. It is clear that the target of the requirements engineering must be done clearly enough for all of the participants. The description of the purpose and goal of the use case description and analysis provides the understanding for the partners of what they are doing, and why and what are they helping to achieve. Continuous communication between ecosystem members is one of the key issues in achieving goals: both the single member's, the collaboration partners' and the ecosystem's. When all the ecosystem members use the same RE method, communication and co-operation is easy and fluent inside the ecosystem.

Despite the many advantages, shortcomings were also identified. Especially, the definition of quality requirements needs further exploration; special skills on quality attributes, e.g. performance, reliability and security, are required and should be present in the innovation and requirements analysis, negotiation and specification phases. For the service innovation, the quality attribute-specific ontologies should be provided for the use of ecosystem members. The methods for the elicitation, analysis, negotiation, trade-off analysis and specification of the non-functional requirements should also be provided with the proper guidance. The knowledge management model of the ecosystem is responsible for providing these ontologies and the methods to be used in each RE phase to achieve the non-functional requirements. Furthermore, the use of these quality ontologies and methods should be supported by the ecosystem support services, e.g. quality specific tools as support services provided via the cloud. Therefore, there should be an arrow between Knowledge management model and ecosystem support service elements in Fig. 2. There already exist approaches that can be utilised here, such as quality ontologies [58], quality-driven design methods and tool support for attaching quality properties for architectural elements [59].

Since the ecosystem is dynamic, it evolves all the time as new members, services and value networks emerge. The knowledge management model should evolve too, adapting to the needs of the ecosystem. Also, new support services should emerge as and when needed. For example, in the case of the first usage of the RE method, a demand was identified for a service that collects the new requirements as they emerge. Since the requirements innovation is continuous inside ecosystem, this kind of new service would enable the service providers to detect easily what kind of services has demand inside ecosystem. In addition, as the ecosystem monitors the quality of its services, it should also provide a matchmaking service for service selection to match the required quality with the provided quality.

6 Conclusions

Digital ecosystems bring out new challenges to service engineering; (i) the business and development environment is highly dynamic; (ii) the needs and demands of service customers are unclear and ever changing; and (iii) heterogeneous

Table 8 The method application

	ICARE	CDC
Description of the digital service ecosystem	Ecosystem of cloud services provided for digital content management, processing and delivery in interactive multi-screen TV services	An open service platform offering open real-time data from several data providers (offering data normalisation, integration and analysis, service hosting, open data APIs, service registries and the platform modules and services to third-party application developers)
Countries and partners	5 countries, 25 partners	4 countries, 7 partners
Roles of the partners in ecosystem	Service providers for content processing and rights management, and for user interaction	Data providers and data brokers
	Cloud IaaS and PaaS providers	Platform providers
	Interactive TV application and service developers	Application developers
		Service providers
		Service brokers
Goal of applying the service RE method	Identifying the requirements for a service framework and platform that would enable the digital service ecosystem to build and offer interactive multi-screen TV services	Extract the high-level user and business requirements for the open real-time data platform to be developed
Application of the service RE method	The use case description template was used to collect the information. The results formed a preliminary set of common services and potential new identified services. The analysis template was used for analysing the use cases, their commonalities and differences and clustering the identified services in service taxonomy. Several iterations were required for merging and refining use cases and representing the results as a set of master use cases that as a whole defined the baseline for service architecture modelling	The questionnaire was directed to potential application developers in the consortium. Each party who planned to create a showcase application on top of the platform defined their application use cases with the given RE document. The platform architects did initial requirement analysis for the platform from a user and business perspective. The technical requirement specification was done based on the results of the first two phases. This specification was used as bases for the architecture and system design definitions
Amount of the identified requirements	238 functional requirements	14 functional requirements
	21 non-functional requirements	3 non-functional requirements
	9 business requirements	2 business requirements
	7 constraints	4 constrains
Service taxonomy: the identified digital service groups	User services/applications	Multi-modal mobility services
	Cloud services	
	Home network services	
	Multi-screen interaction services	
	Infrastructure services	
	ICARE service framework services	
Status of readiness	All use cases are under work. The master use cases contain approximately 50 % of the original identified services. A proof of concept implementation is under work and is estimated to be finalised by 28/2/2015	All use cases are under work. The requirement specification and system design phase started in February 2014. The development phase started in June and will last until October, after which the pilots are made. The project ends 31/12/2014

and non-stop emerging technologies are used, or are available, for service implementation. However, service architects should be able to make decisions about what, why and how to develop digital services that have high business potential, are attracting customers and can effectively be developed, operated and maintained.

This paper introduced a novel approach to defining the requirements of digital services in an ecosystem-based man-

ner. First of all, the approach defined what the digital service ecosystem is and how it differs from other ecosystems. Second, the service engineering process of a digital service ecosystem was outlined, keeping the focus on the requirements engineering of digital services. The service RE method introduced three main phases—service innovation, business analysis and requirements analysis, negotiation and specification—as a continuous and iterative engineering process that starts from business and end-user goals and provides a service taxonomy and a set of master use cases as an outcome. Each service is described with the functional and non-functional requirements, constraints and adaptation rules. The use of the service RE method was exemplified by a second-screen voting service, a work done by the Innovative Cloud Architecture for Real Entertainment (ICARE) ecosystem. Practical experiences of using the service RE method was also collected from the ecosystem members; the method was useful for describing, documenting and communicating the capabilities of the digital services. Especially, the method was useful in the requirements analysis phase, where ecosystem members worked together. However, further exploration is required with quality requirements that need special skills and knowledge on quality characteristics in all service engineering phases.

Acknowledgments This work was carried out at the VTT, Technical Research Centre of Finland, within the ITEA2-ICARE (Innovative Cloud Architecture for Real Entertainment) project. The authors thank the ICARE project members for their feedback and Neusoft for the example use case.

References

1. Chang E, West M (2006) Digital EcoSystems a next generation of collaborative environment. In: Eight international conference on information integration and web-based applications and services, vol 214, pp 3–23
2. Ralyté J (2012) Viewpoints and issues in requirements engineering for services. In: IEEE 36th international conference on computer software and applications workshops, COMPSACW, Izmir, Turkey, pp 341–346
3. Liu L, Yu E, Mei H (2009) Guest editorial: special section on requirements engineering for services. IEEE Trans Serv Comput 2(4):318–319
4. Gu Q, Lago P (2009) Exploring service-oriented system engineering challenges: a systematic literature review. Serv Oriented Comput Appl 3:171–188
5. Bano M, Zowghi D, Ikram N, Niazi M (2014) What makes service oriented requirements engineering challenging? A qualitative study. IET Softw 8(4):154–160
6. Bano M, Zowghi D (2013) Service oriented requirements engineering: practitioner's perspective. In: Service-oriented computing workshops (ICSOC 2012), Shanghai, China, pp 380–392
7. Bano M, Ikram N (2010) Issues and challenges of requirement engineering in service oriented software development. In: Fifth international conference on software engineering advances (ICSEA), Nice, pp 64–69
8. Liu P, Nie G (2009) Research on service ecosystems: state of the art. In: International conference on management and service science, MASS '09, Wuhan, China, pp 1–4
9. Knauss A, Borici A, Knauss E, Damian D (2012) Towards understanding requirements engineering in IT ecosystems. In: International workshop on empirical requirements engineering, Chicago, USA, pp 33–36
10. Zhang J, Fan Y (2010) Current state and research trends on business ecosystem, In: IEEE international conference on service-oriented computing and applications (SOCA), Perth, WA, pp 1–5
11. Li S, Fan Y (2011) Research on the service-oriented business ecosystem (SOBE). In: 3rd International conference on advanced computer control (ICACC), pp 502–505
12. Iansiti M, Levien R (2004) Creating value in your business ecosystem. Harv Bus Rev. http://hbswk.hbs.edu/item/3967.html
13. Bosch J (2009) From software product lines to software ecosystems, In: Proceedings of 13th international software product line conference (SPLC'09), pp 111–119
14. Jansen S, Cusumano M (2012) Defining software ecosystems: a survey of software platforms and business network governance. In: Forth international workshop on software ecosystems, Cambridge, MA, USA
15. Hanssen GK, Dybå T (2012) Theoretical foundations of software ecosystems, In: Proceedings of the forth international workshop on software ecosystems (IWSECO), Cambridge, MA, USA, pp 6–17
16. Riedl C, Böhmann T, Leimeister JM, Krcmar H (2009) A Framework for analysing service ecosystem capabilities to innovate, In: 17th European conference on information systems, Verona, Italy, pp 2097–2108
17. Ruokolainen T (2013) A model-driven approach to service ecosystem engineering. PhD Thesis, University of Helsinki, Department of Computer Science, Helsinki, Finland
18. Blau B, Krämer J, Conte T, van Dinther C (2009) Service value networks. In: IEEE conference on commerce and enterprise computing, Vienna, Austria, pp 194–201
19. Schroth C (2007) The internet of services: Global industrialization of information intensive services. In: 2nd International conference on digital information management (ICDIM'07), Lyon, France, pp 635–642
20. Riedl C, Böhmann T, Rosemann M, Krcmar H (2009) Quality management in service ecosystems. Inf Syst e-Bus Manag 7(2):199–221
21. Wu C, Chang E (2005) A conceptual architecture of distributed web services for service ecosystems. In: 18th International conference on computer applications in industry and engineering (CAINE), Hawaii, pp 209–214
22. Ruokolainen T, Ruohomaa S, Kutvonen L (2011) Solving service ecosystem governance. In: Proceedings of the 15th IEEE international EDOC conference workshops, pp 18–25
23. Chesbrough HW (2011) Bringing open innovation to services. MIT Sloan Manag Rev 52:85–90
24. Khriyenko O (2012) Collaborative service ecosystem-step towards the world of ubiquitous services. In: Proceedings of the IADIS international conference collaborative technologies, Lisbon, Portugal, pp 19–21
25. Pantsar-Syväniemi S, Purhonen A, Ovaska E, Kuusijärvi J, Evesti A (2012) Situation-based and self-adaptive applications for the smart environment. J Ambient Intell Smart Environ 4(6):491–516

26. Ferronato P (2004) DBE architecture requirements, del 2.2: Architecture scope document. D.B.E. Digital Business Ecosystem contract no 507953. http://www.digital-ecosystems.org/
27. Ovaska E, Kuusijärvi J (2014) Piecemeal development of intelligent smart space applications. IEEE Access 2:199–214
28. Kutvonen L, Ruokolainen T, Ruohomaa S, Metso J (2008) Service-oriented middleware for managing inter-enterprise collaborations. In: Global implications of modern enterprise information systems: technologies and applications, ser. advances in enterprise information systems (AEIS), pp 209–241
29. Flores F, Mora M, Álvarez F, Garza L, Durán H (2010) Towards a systematic service-oriented requirements engineering process (S-SoRE). In: ENTERprise information systems, communications in computer and information science, vol 109, pp 111–120
30. Immonen A, Pakkala D (2014) A survey of methods and approaches for reliable dynamic service compositions. Ser Oriented Comput Appl 8(2):129–158
31. Al-Fataftah IA, Issa AA (2012) A systematic review for the latest development in requirement engineering. World Acad Sci Eng Technol 6:691–698
32. Loniewski G, Insfran E, Abrahão S (2010) A systematic review of the use of requirements engineering techniques in model-driven development. Model driven engineering languages and systems. Lecture notes in computer science, vol 6395, pp 213–227
33. Husnain M, Waseem M, Ghayyur SAK (2010) An interrogative review of requirement engineering frameworks. Int J Rev Comput 2:1–8
34. Chesbrough HW, Appleyard MM (2007) Open innovation and strategy. Calif Manag Rev 50:57–76
35. Chan CML (2013) From open data to open data innovation strategies: creating E-services using open government data. In: 46th Hawaii international conference on system sciences, Wailea, HI, USA, pp 1890–1899
36. Stathel S, Finzen J, Riedl C, May N (2008) Service innovation in business value networks. In: 18th International RESER conference, Stuttgart, Germany, pp 288–302
37. Fricker S (2010) Requirements value chains: Stakeholder management and requirements engineering in software ecosystems. Requirements engineering: foundation for software quality. Lecture notes in computer science, vol 6182, pp 60–66
38. Schindlholzer B, Uebernickel F, Brenner W (2011) A method for the management of service innovation projects in mature organizations. Int J Serv Sci Manag Eng Technol 2(4):25–41
39. van Eck P, Wieringa R (2003) Requirements engineering for service-oriented computing: a position paper. In: Proceedings of first international E-services workshop, ICEC 03, Pittsburgh, USA, pp 23–29
40. De la Vara González JL, Díaz JS (2007) Business process-driven requirements engineering: a goal-based approach. In: 8th Workshop on business process modelling, development and support, Trondheim, Norway
41. Cardoso ECS, Vitoria B, Almeida JPA, Guizzardi G (2009) Requirements engineering based on business process models: a case study. In: 13th Enterprise distributed object computing conference workshops, Auckland, pp 320–327
42. Khosravi A, Modiri N (2012) Requirement engineering in service-oriented architecture. In: International conference on networks and information (ICNI 2012), Bangkok, Thailand, pp 101–105
43. Liegl P, Schuster R, Zapletal M, Huemer C, Werthner H, Aigner M et al (2009) A methodology for process based requirements engineering. In: 17th IEEE international requirements engineering conference, Atlanta, pp 193–202
44. Hussein M, Yu J, Han J, Colman A (2012) Scenario-driven development of context-aware adaptive web services. In: 13th International conference on web information systems engineering—WISE 2012. Lecture notes in computer science, vol 7651, pp 228–242
45. Seyff N, Maiden N, Karlsen K, Lockerbie J, Grunbacher P, Graf F et al (2009) Exploring how to use scenarios to discover requirements. Requir Eng 14:91–111
46. Kimita K, Akasaka F, Shimomura Y, Öhrwall Rönnbäck A, Sakao T (2009) Requirement analysis for user-oriented service design. Asian Int J Sci Technol Prod Manuf Eng 2(3):11–23
47. Kett H, Voigt K, Scheithauer G, Cardoso J (2008) Service engineering in business ecosystems. In: Proceedings of the XVIII international RESER conference, Stuttgart, Germany, pp 1–22
48. Wiesner S, Peruzzini M, Doumeingts G, Thoben KD (2012) Requirements engineering for servitization in manufacturing service ecosystems. In: Conference on industrial product service systems, Tokyo, Japan, pp 291–296
49. Ruokolainen T, Kutvonen L (2009) Managing interoperability knowledge in open service ecosystems. In: 13th Enterprise distributed object computing conference workshops, Auckland, New Zealand, pp 203–211
50. Dobson G, Sawyer P (2006) Revisiting ontology-based requirements engineering in the age of the semantic web. in: International seminar on dependable requirements engineering of computerised systems, Halden
51. Castañeda V, Ballejos L, Caliusco L, Galli R (2010) The use of ontologies in requirements engineering. Glob J Res Eng 10(6):2–8
52. Xiang J, Liu L, Qiao W, Yang J (2007) SREM: A service requirements elicitation mechanism based on ontology. In: 31st Annual international computer software and applications conference, pp 196–203
53. Kaiya H, Saeki M (2005) Ontology based requirements analysis: Lightweight semantic processing approach. In: Fifth international conference on quality software (QSIC'05), Melbourne, Australia, pp 223–230
54. Ovaska E, Salmon Cinotti T, Toninelli A (2012) The design principles and practices of interoperable smart spaces. In: Xiaodong L, Yang L (ed) Advanced design approaches to emerging software systems: principles, methodologies and tools. IGI GLobal, pp 18–47
55. ISO/IEC, 2001. ISO/IEC 9126–1 international standard: Software engineering—product quality. Part 1: quality model. International Organization for Standardization, Geneva
56. Immonen A, Palviainen M, Ovaska E (2014) Requirements for open data based business ecosystem. IEEE Access 2:88–103
57. OASIS (2008) Reference architecture for service oriented architecture 1.0
58. Zhou J, Niemelä E, Savolainen P (2007) An integrated QoS-aware service development and management framework. In: Working IEEE/IFIP conference on software architecture, WICSA'07, Mumbai, India, pp 136–145
59. Ovaska E, Evesti A, Henttonen K, Palviainen M, Aho P (2010) Knowledge based quality-driven architecture design and evaluation. Inf Softw Technol 52(6):577–601

Publication III

Requirements of an open data based business ecosystem

IEEE Access,
Vol. 2, pp. 88–103.

IEEE Access
practical innovations : open solutions

Received December 2, 2013, accepted January 3, 2014, date of publication January 31, 2014, date of current version February 18, 2014.

Digital Object Identifier 10.1109/ACCESS.2014.2302872

Requirements of an Open Data Based Business Ecosystem

ANNE IMMONEN[1], MARKO PALVIAINEN[2], AND EILA OVASKA[1]
[1]VTT Technical Research Centre of Finland, P.O. Box 1100, FIN 90571 Oulu, Finland
[2]VTT Technical Research Centre of Finland, P.O. Box 1000, FIN 02044 Espoo, Finland

Corresponding author: A. Immonen (anne.immonen@vtt.fi)

This work was supported by the National Strategic Research Project, Open Data End-user Programming funded by the Finnish Funding Agency for Technology and Innovation and the VTT Technical Research Centre of Finland.

ABSTRACT Emerging opportunities for open data based business have been recognized around the world. Open data can provide new business opportunities for actors that provide data, for actors that consume data, and for actors that develop innovative services and applications around the data. Open data based business requires business models and a collaborative environment—called an ecosystem—to support businesses based on open data, services, and applications. This paper outlines the open data ecosystem (ODE) from the business viewpoint and then defines the requirements of such an ecosystem. The outline and requirements are based on the state-of-the-art knowledge explored from the literature and the state of the practice on data-based business in the industry collected through interviews. The interviews revealed several motives and advantages of the ODE. However, there are also obstacles that should be carefully considered and solved. This paper defines the actors of the ODE and their roles in the ecosystem as well as the business model elements and services that are needed in open data based business. According to the interviews, the interest in open data and open data ecosystems is high at this moment. However, further research work is required to establish and validate the ODE in the near future.

INDEX TERMS Business ecosystem, open data.

I. INTRODUCTION

In the future, an increasing amount of services and applications will be developed based on open data. Open data are data that are freely available to everyone to use and republish as they wish without restrictions of copyrights, patents or other mechanisms of control [1]. The benefits of open data have been widely recognized around the world, and there has been a tendency in many countries to open the data of the public sector.[1] In particular, the data that are collected on tax revenues are obligated to be opened in many countries.

Private companies could also open a part of their own data. In addition to earning direct profits from data sales, the opening of data can provide other benefits, such as new partners, new interests in the company's main products/services, new kinds of business activities and new customers for the product/service as a result of data-based applications. However, opening data requires a big change in a company's business. The lack of knowledge on the benefits of opening data, the lack of business models and the lack of new operation models are the main obstacles that explain why companies are not currently motivated to open their own data [2]. The developers of digital services and applications could greatly benefit from the business opportunities of open data. However, the lack of business ecosystems and business models has been identified as the main obstacle to data utilization in services and applications [2].

A business ecosystem is a dynamic structure of organizations that work together in a specific primary technological platform or core business [3]. In an ecosystem, value is not created in a chain but more in a network of actors. A data-based business ecosystem is formed by organizations that each has their own parts and know-how in the data-based business. The ecosystem's actors affect and are affected by the creation and delivery of the offerings of the other actors. Each actor also has a role in the flows of information, material, money and influence relationships between one another. Existing value chains [4]–[6], business models [7]–[11] and open data communities[2] provide building blocks for the business of open data. However, there is still a need for

[1]http://open-data.europa.eu/en

[2]http://ckan.org

an ecosystem to support both the technical perspective and business perspective of an open data based business.

The main contribution of this paper is to outline the Open Data Ecosystem (ODE) from the business perspective and to define the requirements of the ODE that must support different actors and business that are formed around various kinds of data, services and applications. This paper describes:

- The identified actors of the ODE.
- The services that the ecosystem should provide for the open data based business.
- The business model elements required in the open data based business and the description of how the ODE should support these elements.

The ODE outline is based on the state-of-the-art knowledge on business ecosystems, models and actors, and the state of the practice on data-based business and future visions in the industry on open data based business. This knowledge is based on a thorough literature survey carried out in the spring of 2013. The knowledge of the state of the practice is based on interviews conducted among Finnish companies in the summer of 2013. The ODE concept is novel: to our knowledge, there is no model published on open data ecosystems to date.

This paper is organized as follows. The next section compares our definitions in this work with the state of the art and indicates what parts of existing business models can be used in our work. Section III describes the initial outline for the ODE based on a literature analysis. Section IV describes the implementation and results of the company interviews and specifies a new version of the ODE that is modified according of the requirements collected in the interviews. Section V provides a discussion related to the characteristics of the ODE and the results of the company interviews. Finally, section VI makes concluding remarks.

II. THE STATE OF THE ART OF OPEN DATA BASED BUSINESS

A. TERMINOLOGY

The following terminology is used in this paper:

Raw data – Raw data are data that are produced by observing, monitoring, using questionnaires, etc. but have not yet been processed for any specific purpose.

Information – A refinement and processing of data will produce information from the data. A refinement of data can analyze, align and aggregate data from different physical and digital sources and thereby increase the understanding of the data. Raw data can be refined to increase the understanding of the data. This derived information is sorted for reasoning processes that are able to make decisions on the actions that the applications and services have to take and, moreover, how these actions should be performed.

Knowledge – Knowledge can refer to the theoretical or practical understanding of a subject. It can be implicit or explicit, and it is more or less systematic. Here, knowledge is used in both meanings: theoretical knowledge represents explicit knowledge on the meaning of data. Practical knowledge is implicit and less systematically collected, represented and shared. This knowledge is related to experiences of using (open) data in business.

Application – A combination of digital services that provides the data to the different end-users of data in their preferred representations when and where they are needed.

Service – A digital service that provides additional value for data processing and can for example, support data collection, analysis, sharing and/or representation. A service can mine and extract data from input data and produce relevant data for a particular context or domain.

Service chain – A set of services performing data refinement and processing steps and ultimately making the derived information available to users.

B. VALUE CHAINS OF DATA

Latif et al [6] define a linked data value chain that has four entities: a raw data provider, a linked data provider, a linked data application provider and an end-user. The chain supports multiple sources of data; i.e., the data may be acquired from several data providers and may be provided to several application providers. Kuk and Davies [5] introduce the assembly of complementarities involved in the chain from raw data to data-based services. There are parties that structure the raw data, make the data linkable, analyze or visualize the data, share the data within the source code of software and ultimately allow the developers to innovate services on top of the source code. Poikola et al. [4] define the roles in the open data value chain from the data publishing perspective and the end-user's perspective:

- The data publishing roles include: a Storer to collect and save raw material, a Developer to manage and process raw material, an Aggregator to combine and edit data from different sources, a Harmonizer to standardize and homogenize data from different sources, an Updater to update information, a Publisher to publish the data and a Register to maintain the administration of data resources.
- The data end-user roles include: an Application developer to utilize the data as part of the service, an Interpreter to interpret the data and a User of data-based services, e.g., an individual, company, or organization that uses open data applications and interpretations.

Tammisto et al. [11] have conducted research on the roles of linked-data developers and application developers in a Finnish context. The interviewed companies identified three developer roles: a consultant, a linked data developer and an application developer. The consultant role was seen as an additional source of revenue for the open data companies through consulting the raw data providers about the options and possibilities. Moreover, Chen et al. [12] identify two new roles related to data analytics; Data-as-a-Service (DaaS) providers collect, generate, and aggregate the content (i.e., data), and Analytics-as-a-Service (AaaS) providers deliver analytics services to analytics consumers. In addition, data value chains can include other non-profit roles, which support the finding, publishing and marketing of open

data sources, promoting the use of open data and networking-related data. For example, communities using the CKAN, an open-source data portal platform (http://ckan.org/), provide a huge number of applications and visualization components and libraries for the utilization of open data, as well as a regional open data network, the Helsinki Region Infoshare (HRI) (http://www.hri.fi/fi/), which intends to make relevant data easily available.

C. BUSINESS MODELS OF DATA

The business model must support the value proposition of a company in an ecosystem. According to [10], a business model defines how an enterprise delivers value to customers, entices customers to pay for value, and converts those payments to profit. Baden-Fuller and Morgan [9] summarize some common business models, stating that a business model has several characteristics other than value proposition, such as describing business behavior and organizing the company. The business model must describe how value is captured from the innovation. Traditional business models concentrate on gaining profits by overtaking competitors and keeping strict boundaries around the company. Open data force companies to re-think their business strategies and models because an open data based business cannot be shut down within the boundaries that surround the company. The transformation to an open data based business model requires a great deal of investment and newly assessed business model elements. The open data based business model can be based on the capabilities and features of open innovation strategies, open-source software business models, the business models of cloud services and the business models of analytics.

Open innovation breaks the boundaries around a company in the innovation phase; companies can create ideas by themselves, use external ideas or co-create ideas with other companies or with the actors of other communities. Two open innovation processes in service innovation have been identified by [13]: outside-in and inside-out. In the outside-in process the components of external knowledge and innovation are used in service development, whereas in the inside-out process, a company allows external parties to use its knowledge and innovation components in service development. Accordingly, the outside-in process can be applied by application developers, whereas the inside-out process is more appropriate for data/information providers.

Seven *open business models* identified within the context of open-source software (OSS) [8] can be classified into four categories based on how they capture value [14]: deployment, hybridization, complements and self-service. The deployment category includes support, subscription and professional services/consulting business models, which are similar to the proprietary side of the software industry. The hybridization category includes proprietary extensions and dual-license business models, attempting to attract customers by licensing to familiarize the customers with the product/service. This kind of proceeding could easily be applicable for open data providers as well as application developers. In the complements business model, open source software is provided with the vendor that sells and supports the hardware device or appliance. Finally, in a self-service business model, users with similar needs pool their resources and create applications for the community's needs. The self-service model can be applicable for application developers that do not sell applications but otherwise use them in their business (e.g., in communication between partners or inside an organization). According to [8], the modification of a company's business model is the trade-off between underlying value creation dynamics, IP ownership/license choice, community management, and target market/product categories.

The *cloud business model* is especially applicable in the case of large data sets, when storing, processing and analyzing require a great deal of resources. In the future, it can be assumed that large data applications will be the main driver of widespread cloud adoption [15]. A survey of cloud adopters reveals that decision makers currently implement public cloud applications and platforms mostly for business agility [16]. The same survey revealed that the latest technology and support for mobile workers are increasingly significant factors in the decision to move to cloud applications. The transformation towards the cloud business model has already been researched by [17], which reveals that the transformation affects all elements of the business model mainly due to customer-side characteristics, which include pay-per-use pricing, ubiquitous access and on-demand availability.

Data analysis (e.g., mining, extracting and sorting) has a high potential in business. In addition to being used in business, data has been found to be valuable in information- and knowledge-based management and decision making inside companies, helping in the understanding of the line of business and the market situation at hand [2]. The emerging *analytics business models* include the proprietary model, the shared data model, the shared analytics model, the shared value model, the co-development model and the new business development model [12]. The last model describes how DaaS and AaaS provide opportunities for application developers to create new business. In addition, the shared value model and the co-development model are suitable in open data based business; the end-users and partners create value together, and a set of companies participates in the development.

D. PRICING MODELS

There exist several pricing models that can be used in data- and information-related business. Services and applications can be priced commonly based on *features* [18] or *performance* [19], or the customer is charged a predefined price for customer-tailored services and applications usage [20].

The traditional pricing models, such as the *Value model* [21], *Portfolio pricing* [22] and *Market pricing* [23] are applicable for pricing the following three core elements: data, services and applications. In addition, the *Cost-based model* [19] is applicable when data, services or applications are added to the actual products or

customized according to the customer's needs. The service-related pricing models are applicable to all of the core elements; in *Pay-per-use pricing model* [24], the customer pays only for the data/service/application usage, whereas in the *Subscription model* [24], the client pays a fixed price for a certain time frame. Furthermore, information/Internet-related pricing models are also applicable. For example, the *Flickr multiple revenue stream model* [10] involves collecting subscription fees, charging advertisers for contextual advertising, and receiving sponsorship and revenue-sharing fees from partnerships. The *Freemium model* [10] is a free, limited-functionality version of the product/service offered to attract users, hoping that some users will pay a premium for advanced features, whereas the *Free Trial model* [25] offers a free trial of the service for 14 or 30 days to attract users, after which the users are required to pay to continue using the service.

III. THE FIRST OUTLINE OF THE ODE

The objective of the ODE is to promote open data based business by making the development of open data based applications and services easy and straightforward. The first outline of the ODE defines the core actors and elements of the open data based business. The actors have their own motives and benefits when operating in the value network of data. Each actor represents one or more roles in the ecosystem. We decided to use the existing value chains of data as a starting point for identifying the stakeholders of the ODE. It seems that there are three core elements in the ODE that businesses are formed around: i) data, ii) services and iii) applications. The applications use data and services and produce valuable information and knowledge for the user.

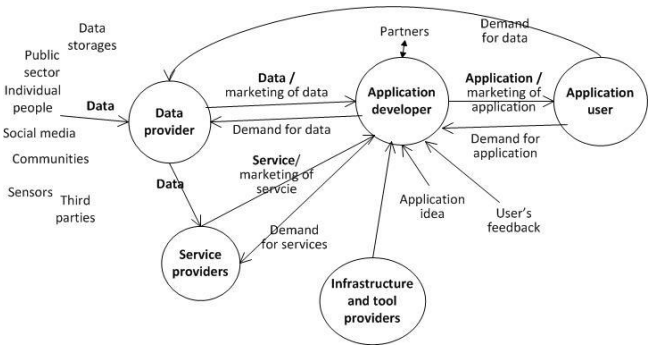

FIGURE 1. Initial actors and their relationships in an open data ecosystem.

A. THE ACTORS OF THE OPEN DATA ECOSYSTEM

There are (at least) five roles for the ODE's actors: i) data providers, ii) service providers, iii) application developers, iv) application users and v) infrastructure and tool providers. Fig. 1 depicts how data flow from the data providers to data consumers. Data providers make data available to other stakeholders, service providers produce services related to data, and application developers use the available data and services and develop applications for the data. Finally, the application users consume the data and services with the help of applications. The infrastructure and tool providers offer other utilities to the actors of the ODE. The following subsections describe the roles of the ODE, the services required for the ODE and the identified data-based business elements in greater detail.

1) DATA PROVIDERS

Data providers are organizations that provide data for the other actors of the ecosystem. The provided data can be raw data, refined data/information or analyzed information. Data providers can be divided into two groups according to their motive: organizations that provide data "for free" without any conditions or with some licenses that restrict the use of data and organizations that that do business from selling access to the data.

Organizations that provide free access to data are usually public administrations or other public entities that have a lot of data but no abilities or resources to use the data in the form of data refinement or the development of services with the data. These organizations provide data to improve the national economy, enabling enterprises and citizens to exploit the data. Several public licenses exist with which a licensor can provide access and copyright permissions to open the data, such as the Creative Commons License and Conformant Licenses. Originally, these licenses granted the "baseline rights" to distribute a copyrighted work without changes at no charge [26]. Most licenses currently contain some license elements that restrict the utilization of data, such as Attribution, Non-Commercial, No-Derivatives and Share Alike. These elements can be mixed and matched to produce a customized license.

The data providers who sell data can attract users to pay for data by providing only a subset of their data as open data, providing guarantees of availability only to paying users, limiting the frequency of access to the open data, providing open access only to stale information, using share-alike licensing, or requiring that people register to access the data [27]. Contracts such as Service-Level Agreements (SLAs) and data fees ensure the quality and permanency of data for the users of the data.

2) SERVICE PROVIDERS

Service providers offer services related to data and can earn income from the usage of services. Those organizations that do not have the abilities and resources to perform the data processing themselves can buy the data processing services from service providers or buy the processed data directly from data providers. Thus, the service provider must: a) identify the needs of customers, b) produce relevant data from input data to a particular context or domain and c) represent the produced data in a usable way. However, it is important to note that a service provider does not necessarily provide a complete service for the user but can simply provide a part of a service chain. These providers may provide ready-made service chains or these service chains may be composed at run-time. In order to make the usage of service chains

easy, these service chains can also be represented as a single service for the user even though the service is composed of several smaller services. In our vision, there are data drivers (described in more detail in [28]–[30]) that make the data and services available to application developers. The data driver services and easy-to-use tools facilitate the application development.

3) APPLICATION DEVELOPERS

Application developers cooperate with partners and innovate applications around open data or use open data as such or integrated with their own data in their applications. An application idea can be provided outside the organization, and the application users' feedback affects the continuous and iterative development of applications. The ODE provides tools and services for developing open data based applications. The applications are created by combining available data and services. Therefore, three kinds of stakeholders participate in application development: application developers create the applications, and data providers and service providers deliver data and services for the applications. Application developers pay for the use of the data driver services and possibly for the use of the data. The revenues are then shared between the application developers, data providers and service providers.

4) APPLICATION USERS

Application users consume data with the help of data-based applications and services. A user can be a consumer, citizen or an enterprise user. A consumer is a user that has bought a commercial application from an application store. A citizen can be user that uses the provided application as a citizen. For example, the application can enable a citizen to produce information from the environment and then consume information provided by the public administration. An enterprise user is a user who uses the applications in business.

5) INFRASTRUCTURE AND TOOL PROVIDERS

Infrastructure and tool providers offer the necessary tools for the ecosystem. The relevant roles are:

- ODE providers who maintain the ecosystem and receive income from the usage of services and applications of the ecosystem.
- Marketplace providers who provide a marketplace in which applications and data driver services can be bought.
- Tool providers who provide tools to develop applications, configure applications for different user needs and execute and control the application. The tools are used as services, and the provider receives income from the tool usage (pay-per-use).
- Cloud service providers who provide the physical facilities for the ecosystem and receive income from the facilities' "rent."

B. THE SERVICES OF THE OPEN DATA ECOSYSTEM

The ODE should provide and deliver the following services to its members:

- Data provider support: The ecosystem has to enable different open data providers to provide data to the ecosystem from heterogeneous data sources.
- Data adaptation: The ecosystem must enable different service providers to develop and provide driver software that is used for adapting the open data to be usable in different applications.
- Tool support: The ecosystem shall provide tools for application developers to create applications, services and new driver software.
- Diverse applications: The ecosystem has to enable and provide application deliveries for diverse application users.

C. THE BUSINESS MODEL ELEMENTS OF THE OPEN DATA ECOSYSTEM

There are several developed business models that are also applicable to data [7], [10], [31], [32]. Firstly, we decided to use the business elements of the Business Model Canvas by Alexander Osterwalder [7] as a basis because it is a well-known model and provides a stable template for developing new business models. The following business elements were identified to be needed in open data based business (summarized in Fig. 2):

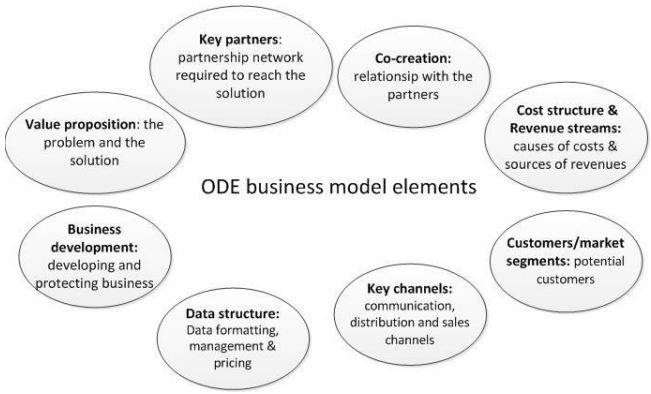

FIGURE 2. Business model elements for open data based business.

Value proposition – This concept deals with the problem and the suggested solution, the value of the solution, identifying a business opportunity based on a demand or innovation and creating a value network or a service chain to reach the solution, identifying the value of the solution to different network actors and identifying the obstacles to reaching the solution.

Key partners – The key partners in the value network include those that provide the solution to the problem and other ecosystem actors that are required to achieve the solution. This concept also deals with the types of relationships with the partners and the benefits/rationale/risks of partnership/cooperation.

TABLE 1. Summary of the interviewed companies.

Company ID	Company role(s)	Company size	Business type	Internationality of business	Usage of data
Company A	Data/service provider	Large	B2C	International	Information enabler products and services
Company B	Application developer, Tool provider	Large	B2B	International	Data-based services
Company C	Application developer	Large	B2B	International	Information-applying services
Company D	Data/service provider, Application developer	Large	B2B, B2C	International	Information enabler products and services
Company E	Data/service provider	Small	B2B	International	Data transfer services
Company F	Application user	Large	B2C, B2B	National	Information-based products, information services
Company G	Data/service provider	Medium	B2C, B2B	National	Information/information-applying services
Company H	Application user, Data/service provider	Small	B2C	National	Data-utilizing services
Company I	Application developer	Small	B2C, B2B	National	Information-applying services, data-utilizing services
Company J	Application user, Data/service provider	Micro	B2C, B2B	National	Information enabler services
Company K	Tool provider	Micro	B2B	National	Information-representing services

Co-creation – This concept involves forming cooperation with partners, identifying the key activities and resources for reaching the solution and the division of the key activities with the partners, and the role of the customer in reaching the solution (e.g., requirements or feedback).

Cost structure and Revenue stream – The main costs of reaching the solution can be calculated after defining the key resources, key activities and key partners. The revenue stream is defined as the way the income is made from each customer segment. The costs and revenue stream are defined for each actor in the value network.

Customer/market segments – These segments include the potential or targeted customers/market segments and the consumers interested in the provided data/information/application. This concept also deals with the type of relationship the company wants to create with its customers.

Key channels– Key channels include communication channels with partners/customers, delivery channels of the solution, and sales channels through which the service/application is delivered/made available to consumers.

Data structure – This concept deals with the uniform format of data, the separation of data between open data and private data, the management of the permissions to data, the management of the ownership of data, the pricing and charges for the use of data, data licenses and license management.

Business development – This concept involves a description of how to develop and protect a business, the definition and management of SLAs, the definition and management of license policies, charges for the use of the solution, the distribution of the revenues with all partners, and the ownership of the solution (IPRs).

IV. THE RESULTS OF INTERVIEWS

The main goal of company interviews was to collect the requirements of the ODE and to receive up-to-date information and views directly from Finnish industries. The theme interview format [33], [34] was selected because of the different backgrounds and contexts of the companies; it would have been difficult to define the exact questions that could be applicable to every interviewee. Each interviewee selected a role that represented his/her company's role in the ecosystem. The roles were defined based on the literature survey described in Section III-A. The interviewee inspected the themes from the viewpoint of this role. In the theme interview, there were no questions, but the themes were informally discussed within the scope of the sub-themes that were relevant to the interviewee.

We defined four main themes with sub-themes:

- Open data (sub-themes: the meaning of data, data as a competitive advantage, data privacy, data integration, data licenses, data sale and user groups): The purpose was to examine how the open data concept is understood and what kind of meaning and role it has in a company's business.
- Application (sub-themes: open data, data drivers, application domain, tool support, IPR, selling/marketing and users): The purpose was to inspect the open data concept from the application viewpoint, i.e., the key issues of open data based applications.
- Co-creation (sub-themes: application idea, partners, division of work, costs and profits, the distribution of profits and contract management): The purpose was to untangle how the cooperation between partners would be formed in the ODE.
- ODE-based business (sub-themes: motives, risks, requirements of the ecosystem, ensuring business and establishing cooperation): The purpose was to find out the opinions and thoughts about the ODE and to gather the requirements of the ecosystem.

Representatives of 11 Finnish companies participated in the interviews. The interviews were performed in Oulu, Helsinki and Tampere between June and August 2013. The companies were selected from different application domains, and interviewees were selected based on their knowledge about the business viewpoints of their companies. Table 1 depicts the backgrounds of the interviewed companies. As can be seen, the interviewed companies differed according to the

TABLE 2. The results of the "open data"-theme interviews.

	Application users	Data/service providers	Application developers	Infrastructure/tool providers
Meaning of data	Competitive advantage	Competitive advantage	Competitive advantage	Beginning of new ideas
Data as competitive advantages	• Several utilization domains • Providing information that competitors do not have • Selling knowledge • Integrating data • Enabling customer to achieve data	• Benefits for early adopters • (Almost) real-time data • Information for a company's core competence	• Rapidly changing and refined data • Market information • Knowledge • Predictions • Technological possibilities	Research data in an early phase about the potential demand for the product
Data privacy	• Private data: cannot be opened • Data utilization requires customers' approval	• The importance of customer analysis • Privacy management	• Private data cannot be shared • Open data are related to public sector	• Data are customers' data • Important how the data are used
Data integration	• Competitive benefit • Provides new information • Standards for content • A need for catalogues and schemas of open data, anonymous open data, and open data consultants		• Open data become private when integrating them with private data • Incompatible data formats • A need for a broker	
Data licenses	• Standards for the representation of open data • Reference to the data source • Data reliability verification	Licenses are required before data can be used	• Clear directives for data utilization • The correctness and permanency of data	• Customers are not willing to pay for information • Data licenses are restricted in contracts
Data sales	• Should consider only for processed data • Information about the customers cannot be sold	• B2B and B2C sales in an ecosystem • Collection of data causes costs	• A small fee for the data • SLA for ensuring the existence of open data sources	
User groups	Application developers, integrators, and end-users		More for hobbies	Research to find out the users' needs

company size, application domain and service type. In addition, a portion of the companies had more than one role in the ecosystem. The size of the companies is defined according to [35]: micro-enterprise <10, small enterprise <50, medium-size enterprise <250 and large enterprise >250 employees. The following subsections introduce the results of the theme interviews.

A. OPEN DATA THEME

Table 2 summarizes the results of the "Open data"-theme interviews.

Meaning of data – The importance of data was identified to be high in all companies, and open data were seen as a trend that companies should follow. Mostly, data were seen as a competitive advantage. Data were also seen as the beginning of new ideas; they help in innovation of new products and services.

Data as a competitive advantage – Information provides a significant competitive edge. The data providers thought that early adopters would benefit from open data. There is a need for consultants that market open data and help companies to understand how open data can improve their businesses. An application user stated that differences are made in services, i.e., what is made with information and how the services are created around information. The customer analysis has a great importance to several companies. Data mining from databases and from data flows and the combination of information will provide many benefits. Application developers did not consider raw data to be important; knowledge and refined information are most important for the business. Generally, almost real-time data was seen to provide a competitive edge. This information can improve safety and enable predictions. According to application developers, rapidly changing information and the refinement of information could create value for customers. In addition, providing information that competitors do not have, selling data-based knowledge and integrating data are seen as competitive benefits. The information can also be produced as a service to the customer. Thus, an application user company sees itself as an information enabler; the company's products/services assist the customer to produce information. A tool provider stated that the business potential and marketing of the products and services are required to be taken into account in the early phase. Market research is required to find out the potential demand for the product. Information for a core competence is achieved through scientific research. Predictions, market information, and technological possibilities also assist in this competition.

Data privacy – Private data are often the customer's own data or information collected about customers. The management of privacy is extremely important to the interviewed companies. For example, companies cannot typically give up or sell information about their customers. The statistics that are collected from the customers or from the customers' data can be delivered further only with the customers' approval. Open data will become private data when private elements are added to the open data set.

Data integration – Data integration is seen as a competitive benefit and provides new business possibilities. Open data can

TABLE 3. The results of the "application"-theme interviews.

	Application users	Data/service providers	Application developers	Infrastructure/tool providers
Open data	• Need for processed data • Need for intermediate data-processing layer	• Raw data • Need for real-time and global data • Need for reliable, secure and permanent data and data sources	• Standard APIs are required • Permanency of data sources • Changing data	
Data drivers	• Unify, analyze and visualize the data • Implementation should be done by third parties • Companies pay for the drivers used	• Implemented by integrators • Data adaptation for users • Need for data mining • Data are in a format that is required by customers • Ensuring data privacy	Handling of privacy	
Application domain		Applications for decision support	• Applications for operators and customers • Need for user-driven development	Not restricted
Tool support	The tools are created by third parties	Ready-made/standard solutions available	Need for standard tools	The tools are created by third parties
IPR		Ensure that the business is under own control	Need for clear IPR	The IPRs belong to a customer
Selling/ marketing	Small markets for applications in Finland	It is difficult to: • Find good applications • Find out what information is available	Applications can be provided as a service	
Users		• Achievement of new user groups • Usability testing; importance of UIs	Need for open data provided in standard formats	

provide additional value for other data sources. For example, the integration of open data with the information gathered by itself creates information that nobody offers at this moment. In addition, it should be possible to integrate one's own information with the advertisers' information. The application developers said that data integration should already be considered in the data-processing phase; the data should be in a form that can be integrated later. There should be schemas for open data. The produced content should follow the related standards. There is also a need for a data broker supporting the publishing of data and sharing of data.

Data licenses – Data processing and the collection of data causes costs, and thus, in many cases, it is not possible to provide data to customers for free. Application developers estimated that there could be a small fee for the used data; the producer will earn money from the data if it ensures the existence of the open data sources and provides some kind of SLA for the open data source. There should be standards for the representation of open data; at least, the government of different countries should obey standards. It should be clear how the data can be used in applications and services. The application users thought that there is an ethic viewpoint in the data licenses; it would be reasonable if the original source of the data would be referred. This assists in the estimation of the reliability of data: the application user could see that a reliable actor has provided the data and nobody has changed them or made their own conclusions as a result of the data. Application developers felt that it is unclear who has the responsibility for the correctness of open data and the permanency of APIs and data sources. According to a tool provider, customers often want everything to be free and do not want to pay for information. Data licenses and other payments are often defined or restricted in contracts. The data providers estimated that licensing rules depend on an application, i.e., how critical the application is. There will be different contracts for different data, e.g., for free data and for certified/audited data. Often, officials provide their own contracts that must be used.

Data sale – The data processing provides additional value for data. For example, processed data could be used for targeting advertisements for certain customers. Different shops and stores could be ready to pay for this kind of information. A data provider estimated that a company's sales in the ecosystem would come from B2B and B2C sales.

User groups – The data providers believed that there are several kinds of users, such as application developers, integrators and end-users for open data. These users' goal is to make all information available to the all actors. A tool provider revealed that the recognition of the user groups of a product/service is sometimes difficult. Discovering the problems and needs of users requires more user research. An application developer estimated that more applications will be required for hobbies in the future.

B. APPLICATION THEME

Table 3 summarizes the results of the "Application"-theme interviews.

Open data – Data providers' content is more or less raw data or information. Service providers can then produce refined information and knowledge from the content. In the future, the goal is to provide more open data for customers. Customers have a use for slowly changing data, but fresh and rapidly changing information is more important. Users can give feedback to data providers and thus improve the

correctness and quality of the data. The interviewees thought that it is more secure to use popular sources of open data in business-critical applications; the reliability and permanency of a popular open data source can be assumed to be better than a less-known open data source.

A data provider emphasized that due to operating a global business, it is not enough that certain (e.g., location-related) data are open in Finland; they should be open everywhere. When data are not open, certain services are only available in certain countries. Application developers emphasize that standard formats are required for open data. In addition, there should be a clear description of the API of an open data source. The data providers and application developers said that there should be a means to ensure the permanency of data sources and APIs and the reliability of data. The risk that data sources disappear can limit the usage of open data in applications. SLAs are needed for open data sources. Additionally, business should not be trusted in the hands of one data provider; there should always be a back-up plan.

Data drivers – Application users need processed data but cannot typically implement the processing itself. Data/service providers identify the need for data-processing services and for an intermediate data-processing layer. Data drivers were considered important, but third parties should implement these drivers because the companies find it difficult to implement the drivers by themselves. A driver should form and unify the data, analyze (or parse) the data and visualize the data. Drivers could be used, for example, for advertisement; the advertising companies could have some kind of link to the driver for which they would pay. The data must always be adapted for the user; they must be in a format that is needed in the customer's core business. The handling of users' privacy should be considered in the data processing of data drivers.

Application domain – Open data can make it possible to provide extra features to applications and will provide new opportunities for software integrators and application developers. The application domain would not be restricted to any specific domain. There could be applications for operators and applications for customers. Applications could also extend to existing big software systems by providing new features. Applications that support decision-making are needed in different application domains. For example, there is a need for applications that guide the user or provide a snapshot of the overall situation. Open data requires data-mining tools to extract the essential data from the data flows. Data security has an important role; there must be ways to recognize users and control the access to data.

Tool support – Tool support was considered self-evident. The tools and applications for processing are created by others. The tools should be standard, and they should have stable and compatible APIs. The challenge presented by these data is that often, situations are unique, and it is difficult to represent the data with the available tools.

IPR – Clear IPR is needed. It is important to make sure that the business is under its own control. In customer work, the IPRs belong to a customer.

Selling/marketing – Application users emphasized that there are small application markets in Finland. Several data providers stated that it is difficult to find out what kind of information is available, and it is difficult to find good applications from app stores. According to application developers, applications are often based on commercial agreements. Applications can be provided as a service or be sold to other operators (B2B). In addition, there are licensed applications, whereas a portion of these applications is a company's own property. Furthermore, applications are not always in the core, but there are markets for integrated solutions consisting of different kinds of devices and software systems.

Users – An application is often a way to provide a service to customers. The usability of applications is increasingly important in competition and easy-to-use and easy-to-learn user interfaces are needed. Thus, there is a great need for the usability testing of applications. In addition, application developers could tailor applications to different consumer groups of open data.

C. CO-CREATION THEME

Table 4 summarizes the results of the ''Co-creation''-theme interviews.

Application idea – The application idea can come from customers or from the company itself. In many cases, ready-made products or services can be applied, and standard APIs assist different stakeholders in integrating their solutions to SW systems.

Partners – Generally, the companies had quite compact and stabilized partner networks. Some companies contracted a great deal of consulting work, and some also use subcontractors in software development. In a project type of work, the customer sometimes selects the partners, and the company sometimes selects the partners. According to a tool provider, the selection of a partner was dependent on the case; the different partners are usually oriented to different technologies.

Division of work – Each actor must have a natural role in the ecosystem; otherwise, actions must be implemented by themselves. The division of work between partners must be strictly defined for each task. A tool provider stated that by cooperating with partners, a company can accept larger projects. Some companies saw themselves as independent actors in the value chain in which there was not much communication between different actors. Some companies saw themselves operating in a value network, as the information and money moved in many directions. The responsibilities should be clearly shared between actors; all actors do not do everything. In addition, clearly defined APIs, components, and the named responsible must be defined for each task.

A data provider company described itself as an actor in a data flow. All data are not necessarily delivered for the next actors in the data flow, but there can also be secondary flows in the data flows. It is important that customers participate in the early stage of the creation of application concepts. Pilot customers can provide feedback for development. One interviewee felt that the more partners there are, the more

TABLE 4. The results of the "co-creation"-theme interviews.

	Application users	Data/service providers	Application developers	Infrastructure/tool providers
Application idea		Importance of standards	The prioritization of application ideas/features	• The application idea comes from the customer • Styles for how to interview people for ideas
Partners	• Compact partner network • Stabilized partners • Consulting work contracted	• Several partners • An actor in a flow of data • Customer participation	Integrates software from sub-contractors	• Project work with customers and partners • Several stabilized partners • Partner selection based on technologies
Division of work	Operation in: • Value networks • Value chains	• Co-creation inside value network • Clear sharing of responsibilities		• Responsibilities are strictly defined • By cooperating, larger projects can be accepted
Costs and profits			Arrangement in each co-creation activity	
Distribution of profits		Mechanisms for agreements	A win-win situation for all stakeholders	
Contract management	Flexible contracts	A new consortium requires new contracts	Open-source solutions should be used in the development	

complicated the things are. However, when the company succeeds, it will be easy to find new partners.

Costs and profits – The earnings/benefits should be shared in the ratio of the work done. The sharing of tasks, costs and revenues is often difficult to agree upon beforehand. These specific assignments can be agreed upon later so that in each co-creation activity, an agreement is made with regard to these issues.

Distribution of profits – According to the data providers and application developers, cooperation does not work if an actor takes too large a share of the profits. It should be a win-win situation for all stakeholders. There should be fair mechanisms of the sharing of profits between partners.

Contract management – The contracts with partners are clear. Sometimes, the contracts must be flexible and to be able to be created quickly. An application user estimated that the contracts between partners will be functional only within each layer.

D. ODE-BASED BUSINESS THEME

Table 5 summarizes the results of the ''ODE-based business''-theme interviews.

Motives – All companies saw several motives and advantages of the ODE. The ODE can support the creation of ideas and the visualization of products and provides a new kind of basic function and new data-based content. Data can provide additional value for products, and the ecosystem enables the creation of more services/applications that do not compete against the company's own products. Thus, open data are seen as a way to improve competitiveness. The opening of data increases the utilization rate of data and makes the data available to a larger number of developers that create applications for special groups and for specific purposes.

Civil data improve the accuracy of the snapshot of the overall situation. However, there must be some kind of reward from produced data, and citizens must benefit from the data. The ecosystem enables an increased understanding of what could or should be done and with whom. The ecosystem also assists in achieving new partners and new customers and enables a company to serve its customers better. For example, the interviewees identified that the customers could use information more effectively in business. The ecosystem could also increase sales and facilitate sales and marketing efforts.

Risks – Open data can change the business environment, and companies also saw some risks in the ODE. Firstly, it is possible that a business would not be profitable in the ODE. Costs increase, but consumers do not use the services/products. Furthermore, there is a large amount of work in terms of the maintenance, support and marketing of the ecosystem. Secondly, because the situations in companies are changing continually, no company seeks to enter into very binding contracts. In this way, it is easy to break contracts. Thirdly, the concept of data drivers was seen as a risk: the application user stated that it would be more rational to use or acquire open data directly from a data provider than to be dependent on the providers of data drivers. There is also a great risk that the data driver service is not available when it is required. Fourthly, for open data, the identified risks included that the data could not be provided openly, the privacy rules change, and the data are so different that a single platform cannot manage all data. The data providers were worried about the quality of data and changes in the quality of data. The quality of data is not in their own hands. Furthermore, the easy access to data can bring new competitors. Open services can cause problems for chargeable services if another actor provides the same service for free. A tool provider was afraid that customers may not understand the concept and use of open data; the customers may be afraid that because the information is free, they must still pay for it according to the contract.

Requirements of an ecosystem – A business ecosystem cannot be created, but it should naturally emerge. There is no ecosystem if the ecosystem does not provide a role and

TABLE 5. The results of the "ODE-based business"-theme interviews.

	Application users	Data/service providers	Application developers	Infrastructure/tool providers
Motives	• New content, ideas and basic functions • Increased understanding in business • Easier service marketing and sales • Improved competitiveness • New customers	• Increased sales • More ecosystem users • Better offerings • Increased amount of application developers • Increased data utilization rate • Willingness of follow the open data trend	• Customers' demand for open data solutions • Improved brand • Available data that provide additional value to the products	• New customers • New partners
Risks	• ODE-based business is not profitable • Availability of data driver services • No binding contracts are made • The data cannot be provided as open	• The ecosystem requires a great amount of work or does not provide a role or benefits for its actors • Lack of open data operators • Low quality of data • New competitors	• Platforms do not work because data are so different • Business requires much work; 24/7	Customers do not understand the open data concept
Requirements of ecosystem	• Support in business know-how and finding new markets • Data validation • Definition of data/information creation • Standardization of data drivers	• Easy and clear joining of the ecosystem • Support for all members • Standard-based data and APIs • Operator services • Data validation	• A developer community for tools, APIs and frameworks • Trust that the framework works in real life	• Support for internationality, market research and contract making • Information about what data there are available
Ensuring business		• Standards • Business models • Replacing actors and solutions with others • Paying for the data quality		
Establishing cooperation	Will be settled practices for cooperation as the operation begins	It will be easier to get partners to join as the size of the ecosystem gets bigger		

benefits for all of its actors. If the ODE is established, it should be a proper investment from the beginning. Joining the ecosystem should be easy and clear. The ecosystem should provide proper support to all of its members and support the development of contracts between actors in the ecosystem. Often, companies are afraid that their ideas may be stolen. The contracts should clearly define the rights of each partner and how payment issues are dealt with. The growth rate of a business can be very fast in the ecosystem. A data provider estimated that there should be more operators of open data. Operator services are needed for a centralized payment mechanism and for a centralized mechanism to share profits inside the ecosystem. The ODE should support in terms of business know-how, finding new markets and assisting companies to see what kinds of products and services are in demand and what consumers really want. Often, Finnish companies let technologies lead the design. They use information but in a misguided way. The result is that they produce a complicated product for which there is no demand. The ODE should also consider internationality; cultural differences should be taken into account. The responsibilities of data verification should be defined in the ecosystem. It seems that there is a need for a data quality verification service in the ecosystem. The ecosystem should provide information about the available data. The ecosystem should also define how the data/information are created. There should be definitions and possibly standards of how to create measurable information, how to store it, etc. Local and unstandardized APIs will create risks for the whole ecosystem. The SLAs for data were also suggested. Furthermore, some form of standardization could facilitate or change the role of the data drivers. In this way, it would be easier to find another service that has the same functionality. There should be a developer community for tools, APIs and frameworks. In addition, there should be a mutual trust that development continues and that the framework works in real life.

Ensuring business – According to the data providers, to ensure business in the ODE, it is important to avoid monopolies. The network must be designed so that it is possible to replace the actors and used solutions with new ones. The usage of standard APIs helps in this regard. New kinds of business models are required when acting in the ecosystem, and the work performed must have value in the business. In addition, by knowing the customer's businesses well, services can be tailored for customers and their information systems. A data provider would be willing to pay to ensure the quality of data sources. Some interviewees saw that there could be an opportunity for the ecosystem and business opportunities for companies that produce processed information. There will also be a demand for data integration and analytics.

Establishing cooperation – As cooperation in an ecosystem begins, there will be settled practices. The size of the ecosystem is important; first, it will be small, but the bigger it gets, the easier it will be to find partners to join it. The larger the ecosystem is, the more interesting it is to application developers.

E. A REFINED OUTLINE OF THE ODE

We modified the first outline of the ODE based on the requirements obtained in the theme interviews. The theme interviews revealed that value networks are formed dynamically among actors to reach a solution. Thus, the ODE should support fast networking; once a demand has been identified, it should be possible to quickly establish a value network, ensuring that different actors can find needed partners, and agree on the division of work, distribution of costs and distribution of profits. Several value networks co-exist inside the ecosystem; they are formed and dissolved according to the situation at hand.

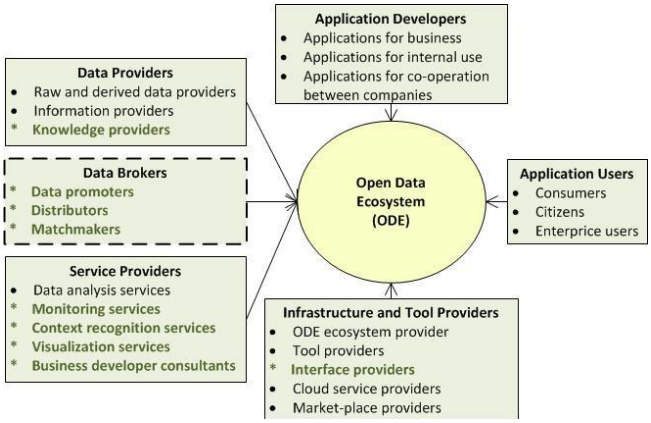

FIGURE 3. **The roles of actors in the refined outline of the ODE.**

The interviewees identified new kinds of actors and services that are needed in the ODE. Fig. 3 introduces the refined outline of the ODE through the actors' roles. A new actor group "Data brokers" is depicted in the box with dashed lines, and new roles within the previously defined actor groups are illustrated with the * mark. The following subsections discuss the identified ecosystem actors, roles and services.

1) NEW ACTORS AND ROLES IDENTIFIED

An entirely new actor class, Data Brokers, was identified. In addition, new roles were identified in the Service Provider, Data Provider and Infrastructure and Tool Provider classes.

Data Brokers–The broker actor includes the following roles:

- Data promoters: A data promoter finds out data and advertises them to the actors (e.g., application developers) of a certain domain or to matchmakers. It also maintains "a list" of available data in the ecosystem and the quality of data, price, applied licenses, etc. The promoter receives a fee from the data providers for the advertisement of their data and from end-users or consumers (when using the data directly) to mediate information about the data source.
- Distributors: These provide the communication channels and distribution channels of data and applications. A promoter receives a fee from the channel user depending on the channel and the usage type (e.g., communication, data transfer).
- Matchmakers: The ecosystem should uncover the demand for data and the quality of the demanded data from end-users and match the demand with the best available data source (received from the data promoter) and data transfer service. It should also inform if there is a demand for not-yet-available data. This matchmaking also enables the automation of data matching and selection in the digital environment.

Service Providers – There is a need for the following new roles of service providers:

- Monitoring service providers: These services are capable of monitoring data in physical and logical environments and detecting any changes in data. Based on monitoring services, the actors can create applications that can automatically or semi-automatically react to the changes (in real time) in the environment and, for example, notify the application users about changes in the environment.
- Context recognition service providers: Monitoring services require services that are capable of recognizing contexts from data streams.
- Visualization service providers: These services are capable of visualizing the essential information in an understandable way to different user groups.
- Business developer consultants: An open data consultant that assists companies in understanding the possibilities of open data in a company's business. It knows what kind of data a company owns and identifies customer needs and the utilization areas of the available data.

Data Providers – There is a need for knowledge providers that offer expert services based on knowledge refined from information.

Infrastructure and Tool Providers – There is a need for interface providers that develop user interfaces for consuming data with different kinds of devices.

2) IDENTIFIED NEW SERVICES

The interviews revealed that the ODE should provide services for:

- Contract-making: The ecosystem should assist the actors in developing contracts with each other. This contract-making should be clear and rapid.
- Finding partners: The ecosystem should maintain a catalogue of the ecosystem's actors and the expertise and reputation of each.
- Finding services: The ecosystem should assist in finding applicable services and candidate services.
- Finding information: The ecosystem should be aware of the available open data/information and assist the actors in finding relevant information.
- Finding markets: The ecosystem should have a means of market research and of finding new markets and customers.

- Data validation, and the definition and standardization of data and interfaces: The ecosystem should define and describe the used syntax and semantics of open data.
- Business models: The ecosystem should assist in defining the appropriate business model for the actors of the ecosystem.

V. DISCUSSION

The results show that an open data based business model can bring both direct and indirect benefits. In open data based business, data flows are created between data producers and data consumers. However, it seems that in many cases, a cash flow between data producers and data consumers is not created, but an open data business can indirectly benefit the business of the producer of the data. One goal can be outsourcing; there could be data providers that provide data for free and hope that there will be actors that develop useful applications and services around the data and will therefore support the business of the data provider. Another option is to sell data with a pay-per-use model or achieve profits in data-based application usage. In addition, a need for free open data and for licensed/premium open data can be identified. Openness is the key work in business models, as an open data based business moves ahead from the proprietary software business. The transformation to the cloud-based business model should also be considered. The cloud-based business model provides solutions to scalability, capacity and interoperability problems and enables customer-side characteristics, such as pay-per-use pricing, ubiquitous access and on-demand availability.

The selection of companies and interviewees affects the results of this research. Thus, the represented results/requirements cannot be generalized for all kinds of companies. For example, the requirements of the ecosystem could have been different for companies whose core business does not involve data/information. In addition, the selection software professionals instead of business professionals could have resulted in more technical outcomes of the interviews. The interviews were performed in Finnish companies. Thus, the results of the interviews cannot be directly globally generalized. However, because several interviewed companies conduct international business, it can be assumed that the results may also be valid in countries other than Finland.

The rest of this section summarizes the results of theme interviews from the perspective of the core elements, open data, services and applications. The following subsections discuss the most important challenges and opportunities that relate to these three perspectives, how the business model elements are supported in the ODE, and the feasibility of the ODE.

A. CHALLENGES AND OPPORTUNITIES OF OPEN DATA

Heterogeneous open data, non-standard APIs and varying licensing conditions complicate the use of open data sources in commercial applications and services. Several data utilization domains were identified in the interviews. In addition, several business opportunities were found, such as providing information that competitors do not have, selling data-based knowledge, integrating data and data mining. There is an increasing need for raw data, refined information, knowledge, rapidly changing information, predictions and market information. The reliability of data and data sources were seen as important in the interviews. Free data are often assumed to have low quality. Furthermore, the permanency of free data sources is considered uncertain. Contracts (e.g., SLAs) were seen as a way of ensuring the existence and quality of open data sources. In addition, there are business opportunities for data promoters validating the data before advertising them further to other ecosystem actors.

It seems that the companies currently do not have much knowledge about open data and the license conditions of open data sources: Firstly, the open data sources should be marketed for companies. For example, public and well-known catalogues are needed for open data. Secondly, the companies should identify the benefits of open data and recognize how open data can be used in their business in the future. There is a need for matchmakers that know what kinds of open data are available and also know what kinds of data company own and identify customer needs and the utilization areas of the available data.

Open data are now often available via non-standard APIs. As a result, there are strong dependencies between an application/service and open data sources and are difficult to create applications/services that work in different regions and in different countries. A standard way to describe data sources and APIs is required as well as a uniform format for the data. Because a large amount of data is private (e.g., the customer's own data, or data about individual persons), the management of data privacy is important. In addition, varying licensing conditions complicate the use of open data sources.

B. CHALLENGES AND OPPORTUNITIES OF THE SERVICES OF OPEN DATA

There is a great need for data processing, analysis, and integration services. The data processing provides additional value for data and the integration of open data with the private data creates information that nobody offers at this moment. Data mining from databases and data flows and the combination of information will provide many benefits. Generally, almost real-time data was seen to provide a competitive edge. The services of real-time data can improve safety and enable predictions. The huge amount of open data and the real-time property brings a challenge to data mining and semantics handling.

There are several technical/non-technical challenges related to the use of open data in business and in internal use. The data integration should be considered in the data-processing phase. The data processing is costly; there is a clear need for the intermediate layer of data processing. In addition, optimizing algorithms are required for the analysis of open data. An observer agent is needed to obtain relevant, new and accurate data for the service

provider. The utilization of silent data requires consideration of how the silent data can be extracted and opened. In decision-making, the exploitation of open data with internal information could assist in process improvement and automate decision-making. Decision-makers need non-numeric information and essential information that must be extracted often from larger data sets. The information should be finally visualized for decision-makers. In the customer interface, personalized digital services can be provided for open data. The challenge remains of how to exploit open data with customer-specific data.

C. CHALLENGES AND OPPORTUNITIES OF THE APPLICATIONS OF OPEN DATA

Applications can be developed for sale, for a company's internal use and for cooperation with partners. The developers of businesses receive incomes from application sale and usage. Developers for an organization's internal use create applications for the organization itself, for certain user groups, or to be used inside organizations, obtaining more efficient productivity and business. Developers for cooperation between partners develop applications to improve the cooperation between partners, obtaining more efficient cooperation with business partners.

The open data enable the creation of new features of the applications. In addition, the application can extend existing software systems. For example, there is a need for decision support and guidance applications. In addition, open data provide new ideas and totally new kinds of user-innovated applications. However, at the same time, the usability of applications must be ensured. Market research is required to find out users' needs. Thus, it is important to collect ideas for applications and feedback from the users and enable the co-creation of applications with different partners. There are small markets for applications in Finland. Thus, it should be possible to adapt applications for different regions and for different countries. For the cross-country open data applications, the standard APIs and globally opened data are prerequisites. A validation of the quality of data was required as well as assurance of the permanency of data sources.

D. HOW BUSINESS MODEL ELEMENTS ARE SUPPORTED IN THE ODE

The ODE supports business model elements as follows:

Value proposition – The ODE services provide assistance in finding what data/information/service is available for innovation and application development.

Key partners – The ODE provide assistance in finding partners, e.g., using a catalogue of classified, registered ecosystem actors. The data promoter ensures the trustworthiness of data providers in the ecosystem.

Co-creation – The ODE provide assistance in terms of clear, flexible and rapid contract-making with partners, e.g., with ready-made templates. These contracts define the responsibilities and earnings of each partner.

Key channels – Distributors provide the communication channels and distribution channels for data and applications. Network operators provide mobile connectivity. A marketplace enables the sales of data driver services and applications.

Cost structure and Revenue stream – The ODE provide assistance in making cost-sharing arrangements in each co-creation activity and mechanisms for agreements. Arrangements for profits-sharing should be made with regard to the ratio of work, enabling a win-win situation for all stakeholders.

Customer/market segments – The ODE services provide a means of market research and of finding new markets and customers, assistance in identifying different user groups, and assistance in conducting user studies, which can be used to find out the needs of certain user groups.

Data structure – Standard formats of open data define the syntax and semantics of open data and interfaces applicable to all data providers. The use of standard data formats guarantees that the data driver services produce output data in an interoperable way. Data reliability is supported by the validation of data through data promoters, which maintain "a list" of the quality of available data. Licenses provide defined and applicable license policies used in the ODE. Licenses are required before data can be used. Therefore, it should be clear to users how the data can be utilized. The ODE defines the practices of how privacy issues are handled and how to protect data from unauthorized users. Data fees specify the defined and agreed-upon fees for the data.

Business development – The ODE provide assistance in market research to find out the demand for data and data-based solutions and to provide information about new data for new ideas, content, products, etc. Matchmakers assist in matching the demand with the best available data source. Consulting services assist in understanding the possibilities for open data in a company's business. The ODE defines clear IPRs for the data, services and applications and also provides information about alternative data sources and services if the selected ones become unreliable or unavailable. The ODE also assists in defining the appropriate business model by providing a template and guidelines for elements of the business model.

E. FEASIBILITY OF THE ODE

In addition to technical challenges, we believe that the most difficult challenge is to obtain enough actors for the ODE. Firstly, there must be enough actors that see the benefits of the ODE and are motivated to actively participate in the development of the ecosystem. Secondly, the ODE should naturally emerge. All of the ecosystem actors are equal; monopolies should not exist. Each ecosystem actor should identify its role in the ecosystem and create the company's business model accordingly. The joining of the ecosystem should be fast and easy through registration. After registration, the actor has access to all data in the ecosystem. The actors in the ecosystem cooperate in value networks, which

are formed dynamically and rapidly to respond to a certain demand. Several networks may emerge simultaneously inside the ecosystem. In the beginning, the ecosystem is assumed to be small, but the bigger it gets, the more willing the companies are to join it.

There are at least two options for the development of the ecosystem: development from scratch or development by extending the existing technical solutions, such as CKAN, enabling straightforward service and application development, thus minimizing the amount of work needed to establish an ecosystem. This development can be started by a company, community or organization by itself, with partners or with other actors, for example, in the same domain. There are also two options for the content of the ecosystem: 1) the ecosystem emerges around a certain domain, being domain-dependent, or 2) there is a ''universal'' ecosystem applicable to all interested parties from different domains, being therefore domain-independent. It is obvious that there is a great amount of work needed to maintain the ecosystem. It must be clear from the beginning who is responsible for the maintenance, support and marketing of the ecosystem. There can be, for example, a separate ecosystem provider, or one of the ecosystem actors (e.g., a large data provider) takes on the role of an ecosystem maintainer.

VI. CONCLUSION

This paper defines an initial outline for the open data ecosystem based on a literature survey and describes the requirements of the ODE based on information collected by interviews performed in 11 Finnish companies. The interviews revealed the state of practice in data-based businesses as well as the future visions of the industry on open data based business. The results of the interviews assisted in refining the refined outline of the ODE and understanding the challenges that data-based business still embodies.

The study revealed several requirements of the open data ecosystem, including the roles of the actors and the required services that must be defined and implemented while establishing the ecosystem. The interviews helped to identify several motives for joining the open data ecosystem. However, there are still some obstacles and risks that have to be taken into account in the development of the ODE. A great deal of work is required for implementing the defined requirements and overcoming the identified obstacles in order to enable profitable business for the actors of the ecosystem. According to the interviews, the interest in open data based business is high. Thus, the ODE could provide great benefits for the involved actors and their businesses through open data and the services and applications around them.

REFERENCES

[1] S. R. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, *DBpedia: A Nucleus for a Web of Open Data* (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, 2007, pp. 722–735.

[2] A. Immonen, M. Palviainen, and E. Ovaska, ''Towards open data based business: Survey on usage of open data in digital services,'' *Int. J. Res. Bus. Technol.*, vol. 4, no. 1, pp. 286–295, 2014.

[3] M. Iansiti and R. Levien, *Creating Value in Your Business Ecosystem*. Boston, MA, USA: Harvard Bus. School Press, Mar. 2004.

[4] A. Poikola, P. Kola, and K. A. Hintikka. (2011). *Public Data—An Introduction to Opening Information Resources, Ministry of Transport and Communications* [Online]. Available: http://www.scribd.com/doc/57392397/Public-Data

[5] G. Kuk and T. Davies, ''The roles of agency and artifacts in assembling open data complementarities,'' in *Proc. 32nd Int. Conf. Inf. Syst.*, Shanghai, China, Dec. 2011, pp. 1–5.

[6] A. Latif, A. Saeed, P. Hoefler, and A. Stocker, ''The linked data value chain: A lightweight model for business engineers,'' in *Proc. 5th Int. Conf. Semantic Syst.*, 2009, pp. 568–575.

[7] A. Osterwalder, C. Parent, and Y. Pigneur, ''Setting up an ontology of business models,'' in *Proc. 16th Int. Conf. Adv. Inf. Syst. Eng. Workshops*, 2004, pp. 319–324.

[8] J. Perr, P. Sullivan, and M. M. Appleyard, *Open for Business: Emerging Business Models for Open Source Software Companies, Working Paper, Lab2Market*. Portland, OR, USA: Portland State Univ., 2006.

[9] C. Baden-Fuller and M. S. Morgan, ''Business models as models,'' *Long Range Planning*, vol. 43, no. 1, pp. 156–171, 2010.

[10] D. J. Teece, ''Business models, business strategy, and innovation,'' *Long Range Planning*, vol. 43, nos. 2–3, pp. 172–194, 2010.

[11] Y. Tammisto and J. Lindman, ''Open data business models,'' in *Proc. 34th Inf. Syst. Seminar*, Turku, Finland, 2011, pp. 762–777.

[12] Y. Chen, J. Kreulen, M. Campbell, and C. Abrams, ''Analytics ecosystem transformation: A force for business model innovation,'' in *Proc. Annu. SRII Global Conf.*, San Jose, CA, USA, 2011, pp. 11–20.

[13] C. M. L. Chan, ''From open data to open data innovation strategies: Creating E-services using open government data,'' in *Proc. 46th HICSS*, Wailea, HI, USA, 2013, pp. 1890–1899.

[14] H. W. Chesbrough and M. M. Appleyard, ''Open innovation and strategy,'' *California Manag. Rev.*, vol. 50, pp. 57–76, Nov. 2007.

[15] H. Liu, ''Big data drives cloud adoption in enterprise,'' *IEEE Internet Comput.*, vol. 17, no. 4, pp. 68–71, Jul./Aug. 2013.

[16] B. Narasimhan and R. Nichols, ''State of cloud applications and platforms: The cloud adopters view,'' *Computer*, vol. 3, pp. 24–28, Mar. 2011.

[17] J. Myllykoski and P. Ahokangas, *Transformation Towards a Cloud Business Model, Discussion*, New York, NY, USA: Commun. Cloud Softw., Apr. 2010.

[18] G. Tao, L. Yi-Jun, G. Jing, and G. Long, ''Research on the economic features and pricing of digital products,'' in *Proc. ICMSE*, 2006, pp. 152–156.

[19] R. Harmon, D. Raffo, and S. Faulk, ''Value-based pricing for new software products: Strategy insights for developers,'' in *Proc. Portland Int. Conf. Manag. Eng. Technol.*, 2004, pp. 1–24.

[20] A. Sundararajan, ''Nonlinear pricing of information goods,'' *Manag. Sci.*, vol. 50, no. 12, pp. 1660–1673, 2004.

[21] V. Allee, ''Value network analysis and value conversion of tangible and intangible assets,'' *J. Intell. Capital*, vol. 9, no. 1, pp. 5–24, 2008.

[22] L. Pastor, ''Portfolio selection and asset pricing models,'' *J. Finance*, vol. 1, pp. 179–233, Feb. 2000.

[23] V. Abhishek, I. A. Kash, and P. Key, ''Fixed and market pricing for cloud services,'' in *Proc. IEEE Conf. Comput. Commun. Workshops*, Orlando, FL, USA, Jan. 2012, pp. 157–162.

[24] C. Weinhardt, A. Anandasivam, B. Blau, and J. Stosser, ''Business models in the service world,'' *IT Prof.*, vol. 11, no. 2, pp. 28–33, 2009.

[25] A. Gaudeul, ''Software marketing on the Internet: The use of samples and repositories,'' *Econ. Innov. New Technol.*, vol. 19, no. 3, pp. 259–281, 2010.

[26] (2013, Nov. 21). *Creative Commons, Baseline rights, Creative Commons* [Online]. Available: http://wiki.creativecommons.org/Baseline_Rights

[27] (2013, Nov. 21). *Open Data Institute, What is Open Data? Online Guide. Open Data Institute* [Online]. Available: http://theodi.org/guides/what-open-data

[28] M. Palviainen, J. Kuusijärvi, and E. Ovaska, ''Semi-automatic end-user programming approach for smart space application development,'' *Pervasive Mobile Comput.*, vol. 2, pp. 1–32, May 2013.

[29] M. Palviainen, J. Kuusijärvi, and E. Ovaska, ''Framework for end-user programming of cross-smart space applications,'' *Sensors*, vol. 12, no. 11, pp. 14442–14466, 2012.

[30] M. Palviainen, J. Kuusijärvi, and E. Ovaska, ''Architecture for end-user programming of cross-smart space applications,'' in *Proc. 4rd Int. Workshop Sensor Netw. Ambient Intell.*, Lugano, Switzerland, Mar. 2012, pp. 823–824.

[31] H. Chesbrough and R. S. Rosenbloom, "The role of the business model in capturing value from innovation: Evidence from Xerox Corporation's technology spin-off companies," *Ind. Corporate Change*, vol. 11, no. 3, pp. 529–555, 2002.
[32] C. Zott and R. Amit, "Business model design: An activity system perspective," *Long Range Planning*, vol. 43, nos. 2–3, pp. 216–226, 2010.
[33] C. Livesey. (2013, Nov. 21). *Sociological Research Skills: Focused (Semi-Structured) Interviews, Sociology Central* [Online]. Available at: http://www.sociology.org.uk/methfi.pdf
[34] S. Hirsjärvi and H. Hurme, *Tutkimushaastattelu: Teemahaastattelun teoria ja käytäntö (in finnish)*. Helsinki, Finland: Yliopistopaino, Univ., 2001.
[35] European Union Commission, "Commission recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises," *Off. J. Eur. Union*, vol. 46, pp. 36–41, May 2003.

ANNE IMMONEN received the M.Sc. degree from the University of Oulu in 2002. Since 2002, she has been a Research Scientist with the VTT Technical Research Centre of Finland. Her main research interests include reliable service engineering, quality modeling and the analysis of composite services, and quality ontologies. Her current research interests include business models and ecosystems in data-based application development.

MARKO PALVIAINEN received the M.Sc. degree from the Lappeenranta University of Technology and the Ph.D. degree in computer science from the Tampere University of Technology in 1998 and 2007, respectively. Since 1999, he has been a Research Scientist with the VTT Technical Research Centre of Finland. His current areas of research interests include mobile applications and application development methods, ontology-driven software engineering, and the evaluation methods of parallel applications.

EILA OVASKA received the Ph.D. degree from the University of Oulu in 2000. Before 2000, she was a Software Engineer, a Senior Research Scientist, and the Leader with the Software Architectures Group, VTT Technical Research Centre of Finland. Since 2001, she has been a Professor with VTT and an Adjunct Professor with the University of Oulu. Her current area of interest is service architectures, particularly in self-adaptive digital service systems and services. She has acted as a workshop and conference organizer and as a reviewer for scientific journals and conferences. She has co-authored over 130 scientific publications. She is a member of the IEEE Computer Science.

Publication IV

Evaluating the quality of social media data in big data architecture

IEEE Access,
Vol. 3, pp. 2028–2043.

IEEE Access
The journal for rapid open access publishing

Received September 2, 2015, accepted September 23, 2015, date of publication October 16, 2015, date of current version November 5, 2015.

Digital Object Identifier 10.1109/ACCESS.2015.2490723

Evaluating the Quality of Social Media Data in Big Data Architecture

ANNE IMMONEN, PEKKA PÄÄKKÖNEN, AND EILA OVASKA
VTT Technical Research Centre of Finland, Oulu 90571, Finland

Corresponding author: A. Immonen (anne.immonen@vtt.fi)

This work was supported by Tekes and VTT through the DIGILE's Need for Speed Program.

ABSTRACT The use of freely available online data is rapidly increasing, as companies have detected the possibilities and the value of these data in their businesses. In particular, data from social media are seen as interesting as they can, when properly treated, assist in achieving customer insight into business decision making. However, the unstructured and uncertain nature of this kind of big data presents a new kind of challenge: how to evaluate the quality of data and manage the value of data within a big data architecture? This paper contributes to addressing this challenge by introducing a new architectural solution to evaluate and manage the quality of social media data in each processing phase of the big data pipeline. The proposed solution improves business decision making by providing real-time, validated data for the user. The solution is validated with an industrial case example, in which the customer insight is extracted from social media data in order to determine the customer satisfaction regarding the quality of a product.

INDEX TERMS Architecture, big data, metadata, quality attribute, quality of data.

I. INTRODUCTION

Nowadays there is a lot of freely accessible data available online. This data is made available by different parties, such as public sectors, private companies, different organizations and institutes, single individuals and the different forms of social media. As the amount of data is enormous, the term 'big data' becomes apparent, meaning a massive volume of structured and/or unstructured data being too difficult to process using traditional database and software techniques. Benefits of open data [1] have already been discovered widely around the world. Several public sectors and even private companies have been interested in opening their data, as data exploitation has been recognized to include several benefits for businesses [2]. Recently, also social media data, such as data from Twitter and Facebook, has increasingly interested companies in their business decision making, as these free-formed discussions can provide insight into consumers' opinions, preferences and requirements considering the company or its products/services [3]–[5]. Big Data Initiatives already exist, spreading out in all directions and comprising various themes, tending to end up in innovative economic development. For example, there are political initiatives, like Big Data – Big Deal,[1] promoted by the Whitehouse. A European initiative[2] by the Big Data Value Association focuses on creating value of big data, whereas NIST[3] and researchers in computer science advanced education in India[4] introduce R&D Initiatives. The terms of 'open data' and 'big data' have been familiar concepts also for many companies for several years. At this moment, a new challenge for companies is to develop a business model around these concepts and create new value from the data through large-scale analytics [6]. The big data dimensions; volume, variety, velocity and veracity [6], pose challenges not only to data analytics, but also to the big data systems that must manage all the data.

As a lot of freely accessible data is commonly unstructured or not more than semi-structured [7], [8] and originates from indeterminate sources, the quality and trustworthiness of the data become key issues. Data quality can be defined according to [9]; data that are fit for use by data consumers. Trustworthiness of data has a broader meaning, defining the perceived likelihood that a piece of information will preserve a user's trust in it [10], and consisting of factors that influence how data-users make decisions regarding the trust

[1] https://www.whitehouse.gov/blog/2012/03/29/big-data-big-deal

[2] http://www.bdva.eu/

[3] http://www.nist.gov/itl/ssd/is/upload/NIST-BD-Platforms-05-Big-Data-Wactlar-slides.pdf

[4] http://drona.csa.iisc.ernet.in/~bigdata/

in information. The data-users (in this case, the companies) need to ensure the quality and trustworthiness of data and be able to trust in it in their businesses. At first, when collecting data, the user wants to ensure the reliability of the data and the data source, leaving out suspicious data. Secondly, when further processing and analyzing the data, the user wants to ensure that the quality and relevancy of data are appropriate for the specific situation. Reliable and valuable data enhances business decision making in several ways, enabling, for example, real-time demand predictions, the estimation of trends, and innovation of potential new products/services. The usage of unreliable data, such as data from suspicious sources, or corrupted, subjective, inaccurate or incomplete data, has a high risk for a company's business, and may lead to poor or incorrect business decisions. Furthermore, the usage of valueless and irrelevant data for certain situations causes a lot of unnecessary effort and expenses for companies.

The evaluation of data quality has relevance in one or more data processing phase(s) of big data architecture (i.e. big data pipeline); in data extraction, data processing and analysis, and finally in decision making. Therefore, quality evaluation of big data must be considered during architecture design, when designing how the data goes through the pipeline of a big data system. Difficulties in quality evaluation are determined by the fact that data quality cannot be judged without considering the context at hand [10]; the same quality attribute is applicable to different situations but the evaluation metric is different. In addition, there are no agreed definitions of quality attributes or classification of their applicability to certain contexts. Furthermore, the characteristics of big data, [6], [11], and [12] as such, set special challenges for quality evaluation. The growing amount of semi-structured and unstructured data, new ways of delivering information and user's changed expectations and perceptions of data quality have been recognized as new challenges in data quality research [8]. Thus, it is obvious that new means are required for data quality evaluation for such kinds of big data.

The purpose of this paper is to describe how to ensure the quality and trustworthiness of social media data for company's business decision making. We introduce a novel solution for data evaluation, in which the data consumer can select the applicable quality attributes and evaluation metrics for the context and situation at hand, and evaluate the quality attributes with evaluation metrics. The solution follows the pipeline of the big data reference architecture of [7].

This paper is organized according to the following: Section 2 defines the basic terms used in this work, and provides state-of-the-art of the big data architectures, and the application of metadata, quality attributes, quality metrics and quality policies in business usage. Section 3 introduces our solution for data quality evaluation in big data architecture. Section 4 provides a case example of how the developments are used in practice; an industrial case company achieves insight into customer needs utilizing social media data. Section 5 provides the validation of the trial usage of the solution and identifies the shortcomings and development targets. Finally, section 6 concludes the work.

II. BACKGROUND

A. TERMINOLOGY

The following terminology is used in this paper:

Data – Data that is produced by observing, monitoring, or using questionnaires, but has not yet been processed for any specific purpose.

Big data – Data that is numerous, cannot be categorized into regular relational databases, and is generated, captured, and processed rapidly [11].

Big data architecture – An architecture that provides the framework for reasoning with all forms of data [13]. Thus, it is a logical structure of core elements used to store, access and manage the big data.

Information – Data that is refined and processed for assigning meaning to the data [14].

Knowledge – Understanding of a subject. Knowledge can be implicit or explicit, and it is more or less systematic. Theoretical knowledge represents explicit knowledge on the meaning of data. Practical knowledge is implicit and less systematically collected, represented and shared.

Service – A digital service that provides additional value for data processing and can, for example, support data collection, analysis, sharing and/or representation [2].

Quality attribute – A representation of a single aspect or construct of a quality [9].

Quality metric – A measure of certain properties of the quality attribute, evaluating the degree of presence of the quality attribute [15].

Quality assessment – Assessing of the quality of raw data as such, without considering the context or the intended use of data.

Quality evaluation – Evaluating the quality of information, taking into account the context and the intended use of information.

Metadata – Structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource [16]. Quality metadata describes the quality attributes of the data and the metrics for each quality attribute.

Quality policy – A policy is a collection of alternative tasks and rules, each of them representing a requirement, capability, or other property of behavior [17]. Quality policies are used to generate quality objectives, serving also as a general framework for action [18].

B. BIG DATA ARCHITECTURES

Big data can be categorized according to data sources, content format, data stores, data staging and data processing [11]. Each of these categories represents several new challenges to data-intensive systems. To achieve high performance, availability and scalability, the big data systems are often distributed. Both software and data architecture must be resilient; the data must be replicated to ensure

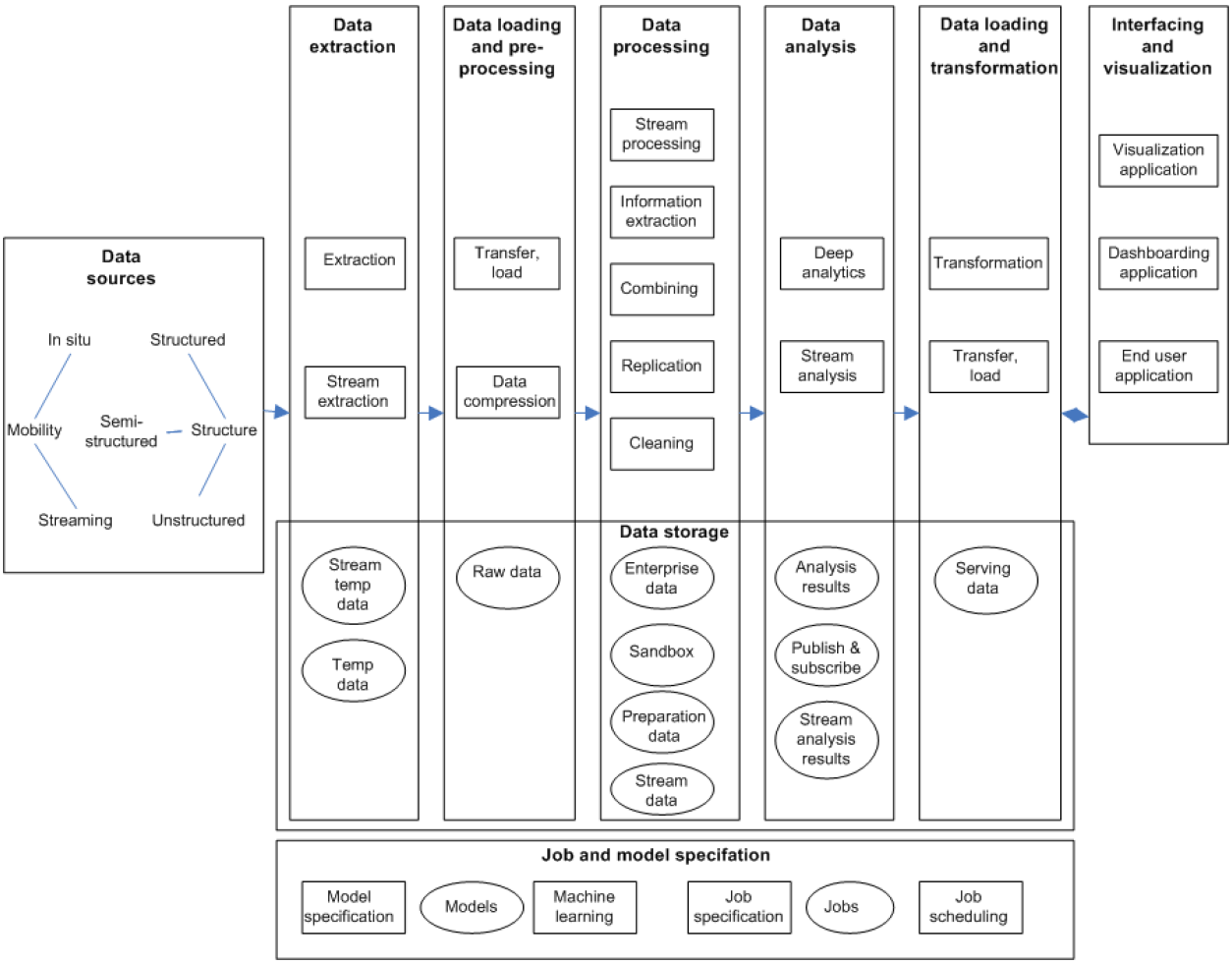

FIGURE 1. High-level view of big data reference architecture [7].

availability and the components of the architecture must be stateless, replicated and failure tolerant [19].

Several implementation architectures of big data systems have been published based on commercial services (Facebook, Twitter, LinkedIn, Netflix, etc.). Recently, a big data reference architecture [7] was published, which had been missing from earlier literature. The big data reference architecture is based on the analysis of published implementation architectures of big data systems. Fig. 1 describes the high-level design of the reference architecture (see [7] for a detailed description and related technologies) derived from published big data use cases. The architecture consists of functionalities (depicted with a rectangle), data stores (circles), and data flows (arrows) between them. Data flows typically from left to right in a big data pipeline. In a big data system, data may be extracted from different sources and stored in a temporary data store. Data may also be further loaded and transmitted into a raw data store, and processed for extraction of new information (to be stored into an enterprise data store). Further, the gathered data is typically analyzed, and results are stored (into a data analysis store). Finally, the analyzed results may be further transformed for serving applications and visualization purposes.

The reference architecture does not consider metadata aspects of big data, which are focused on in this paper.

C. METADATA AND METADATA STANDARDS

The properties of data, such as provenance, quality, and technical details, can be described in metadata of the data, which is simply 'data about data'. Thus, metadata assists end-users to validate the quality and value of data for business usage. However, at this moment the end-users are only slightly satisfied with the metadata available to them [20], and the recent metadata standards do not assist in finding out the quality of data from the data end-user's viewpoint.

Metadata is commonly classified in three categories: descriptive, structural, and administrative metadata [16]. Descriptive metadata identifies a resource and describes its intellectual content. Structural metadata indicates how compound objects are put together, supporting the intended presentation, and use and navigation of a data object. Administrative metadata provides information necessary to

allow a repository to manage objects, such as when, how and by whom a resource was created and how it can be accessed. Metadata standards intend to establish a common understanding of the meaning or semantics of the data. A lot of work has been done by international standardization bodies on standardizing metadata and registries [16], [21]. Data exchange between systems is accomplished by using architectural principles of computer and software systems. The Common Warehouse Metamodel (CWM) [22] is a de-facto standard for data integration by specifying metadata for different kinds of objects found in a data warehousing environment. ISO/IEC 11179 [23] is a standard for metadata-driven exchange of data in a heterogeneous environment, defining metadata and activities needed to manage data elements in a registry. Moreover, the Dublin Core metadata element set enables service creators to describe their own Web resources [24].

A study among data end-users reveals that the end-users consider data quality metadata to be the most useful in metadata [20]. Although several metadata standards exist, it is difficult to estimate their advantages and choose the most applicable one. Furthermore, the standards do not consider data quality aspects from the data users' viewpoint. A data-user metadata taxonomy suggested by [20] facilitates the understanding of various information resources. The taxonomy includes four classes:

- Definitional metadata describes the meaning of data from a business perspective.
- Data quality metadata describes the quality of data when using it for a specific purpose.
- Navigational metadata helps users find the desired data.
- Lineage metadata describes the original source of data and the actions on the data.

D. QUALITY ATTRIBUTES AND METRICS

Several classifications of data quality attributes exist in the literature, but although almost 200 terms for data quality exist, there is no agreement regarding their nature. Some of the quality attributes are too abstract and lack agreed upon specifications for concepts and/or metrics for their evaluation. A lot of work has been done in standardizing quality attributes in the field of software engineering [15], [25], [26]. Quality has also been taken into account systematically in many works dealing with software architecture design [19], [27]–[31]. However, in the case of data, quality issues are not commonly brought into use. Data quality attributes have traditionally been classified into four dimensions important to data consumers [9]. The intrinsic dimension denotes that data have quality in their own right that is independent of the user's context. The contextual dimension considers quality within the context of the task at hand and the subjective preferences of the user. The representational dimension captures aspects relating to information representation, whereas the accessibility dimension captures aspects involved in accessing information. Several other works on data quality and trustworthiness attributes exist, such as [32]–[35], some of them even focusing on social media [36], [37]. The recent research on the quality of online data has been reviewed and summarized under three main factors [10];

- Provenance factors refer to the source of information.
- Quality factors concentrate on factors that reflect how an information object fits for use.
- Trustworthiness factors influence how end-users make decisions regarding the trust of information.

The quality metrics are often designed in an ad-hoc manner to fit a specific assessment situation [38]. Quality assessment metrics can be classified into three categories according to the type of information that is used as quality indicator [38]. Content-based metrics use information to be assessed per se as quality indicator, whereas context-based metrics employ meta-information about the information content and the circumstances in which information was created or used as quality indicator. Rating-based metrics rely on explicit ratings about information itself, information sources, or information providers.

E. QUALITY POLICIES

The quality policy defines which quality attributes are relevant in the context of the task at hand, which quality metric should be used to evaluate the defined quality attributes, and how the evaluation results should be compiled into an overall decision of whether to accept or reject information [9]. A company's organizational policy describes the principles and guidelines required to effectively manage and exploit data/information resources, whereas decision making policy is required for configuring quality evaluation according to the needs of the data-consumer.

The importance of quality policies has been recognized in several works. The Information Quality Assessment Framework [39] enables information consumers to apply a wide range of policies to filter information. The filtering policy consists of a set of metrics for evaluating the relevant quality dimensions, and a decision function that aggregates the resulting evaluation scores into an overall decision on whether information satisfies the information consumer's quality requirements. The approach described in [40] uses an information source filter for subscribing to a set of known information sources, and a scoring function to capture the provenance factors of interest and to assign scores to messages for each factor. The decision making policy allows the decision maker to amplify or attenuate one or more provenance factors that may appear to be more or less important in a particular situation. The framework proposed in [41] uses policies to specify the confidence level required for use of certain data in certain tasks, consisting of three major components: trustworthiness assessment, query and policy evaluation, and data quality management.

Although some promising policy-based approaches already exist for quality evaluation [39]–[41], their practical application is missing. In this work, the represented data quality evaluation solution applies the quality policies.

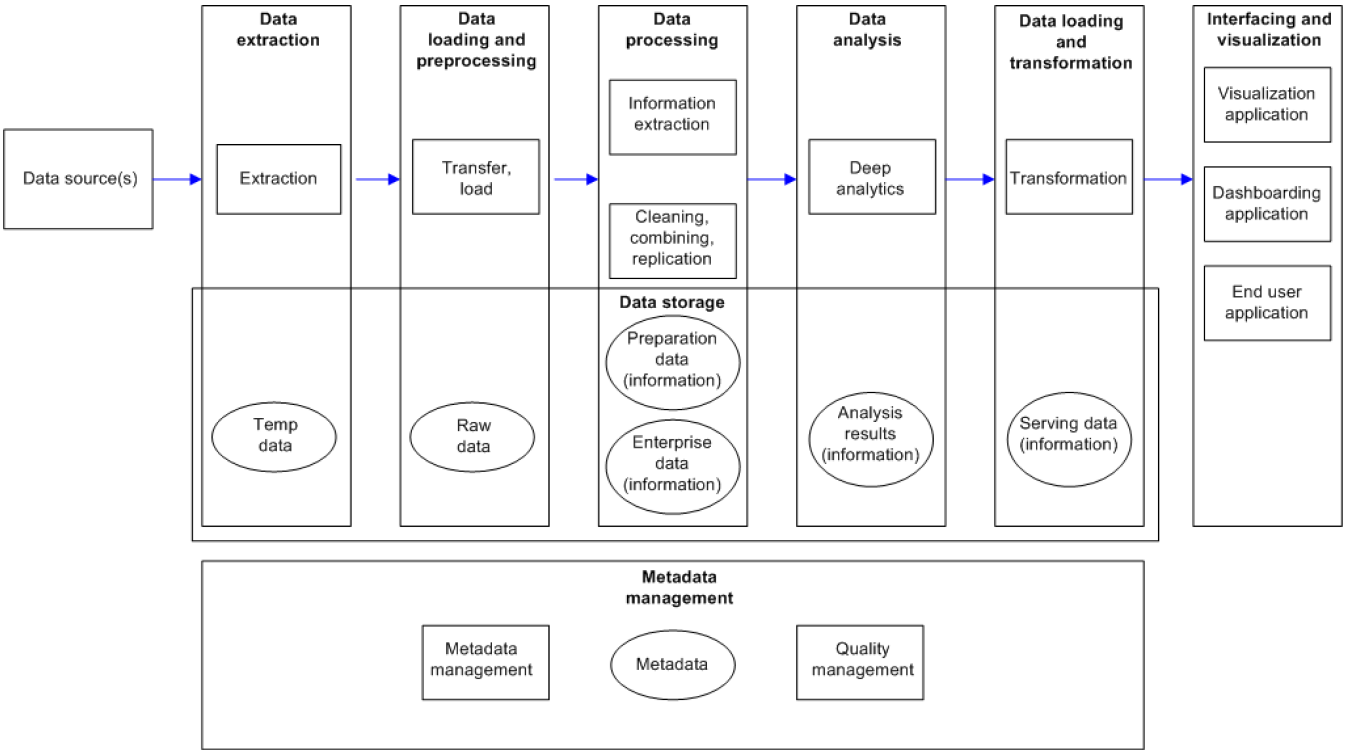

FIGURE 2. Metadata management in big data architecture (enhanced from Fig 1).

III. QUALITY EVALUATION IN BIG DATA ARCHITECTURE

The main purpose of our solution is to evaluate the quality and trustworthiness of data, and incorporate the valuable analyzed results of the data into a company's business decision making process. Data evaluation is conducted in several data processing phases of the big data architecture, going through the pipeline of a big data system. The elements and main phases of the approach are described in the following sub-sections. The metadata in our solution consists of several metadata groups; the whole metadata set is described in the next subsection, but since our focus is on quality, the rest of the paper concentrates only on the quality viewpoint.

Our solution utilizes the big data reference architecture of [7], adding the metadata management element into the big data pipeline (Fig. 2). The metadata management consists of one data store; metadata, and two functionalities: metadata management and quality management. 'Metadata' is a data storage used to store, organize and manage the metadata. 'Metadata management' enables extraction of metadata, and access to metadata. 'Quality management' assigns values to quality attributes based on the properties of associated metadata and data sets.

A. DATA AND METADATA IN THE BIG DATA PIPELINE

1) DATA AND DATA REFINEMENT

Fig. 2 describes the flow of data and creation of information through the big data pipeline. The data that is extracted into a big data system may be structured, semi-structured, or unstructured. Structured data has a strict data model (e.g. based on a database schema). Semi-structured data is not raw data or strictly typed, but instead it has an evolving data model (e.g. JSON/XML documents) [7]. Unstructured data is not associated with a data model, and can have miscellaneous content, such as documents, pictures, videos, etc. Data is extracted from the data source to a company's system as a data set that is an identified collection of data that contains individual data units organized in a specific way and collected for a specific purpose. Extracted data may be stored temporarily (into temp data storage), until it is loaded and/or preprocessed, and stored permanently to raw data storage.

When the data is processed, i.e. cleaned, replicated, combined or compressed, the raw data is transformed to enhanced data, and stored temporarily into preparation data storage. New information may also be extracted from raw data, and saved into enterprise data storage (by storing raw data in a structured format [7]). Deep analytics creates additional insight based on data/information, and entirely new data sets may be created in the form of analysis results. The analysis enables getting value from data and increasing data consumer's understanding of the data; thus transforming the data into information. Data transformation finally modifies analysis results for serving end-user applications (e.g. servicing of analytical queries).

2) METADATA GROUPS

In our approach, metadata is defined as data about gathered data sets in a big data pipeline. The metadata of the data set

TABLE 1. Quality attributes of quality metadata.

Quality attribute	Description and rationale
Accuracy	The degree of correctness and precision. Ensures that the data/information is error-free and the value is in consistent form in accordance with the business data model.
Believability	The extent to which information is regarded as true and credible. When the identity of the informer is known, the information is assumed to be more reliable, traceable, and less likely malicious.
Completeness	The degree to which data/information is not missing. Verifies that the data/information is sufficient in breadth, depth and scope.
Consistency	Implies that two or more values do not conflict with each other. Ensures internal validity.
Corroboration	The same data comes from different sources. Freely available online data can be assumed to be true when the same data comes from several different sources.
Coverage/ amount of data	The extent to which the volume of data is appropriate for the task at hand (appropriate volume of data available). This means that information is of sufficient breadth and depth for the task of the information consumer. The coverage (breadth and depth) can be assessed for each data set, and the large amount of data sets provides assurance in decision making
Validity	Indicates the likelihood that the information is valid in a certain situation.
Popularity	The source provides accurate information, having a number of followers, or the information is liked and therefore repeated by others. The widespread use of a resource tends to lead to more trust.
Relevancy	The extent to which information is applicable and helpful for the task at hand. Non-relevant data sets should not be considered further.
Timeliness	The freshness of the data; timestamp is important for extracted, processed and analyzed data sets.
Verifiability	The degree and ease with which the data/information can be checked for correctness. The traceability and provability of data/information; the data can be verified by users, for example, by using the references to original sources.

is divided into five groups based on the existing standards (e.g. [23] and [24]);

1. Navigational metadata (i.e. where the data set can be found) provides the list of semantic tags or keywords identifying the data set, and the location where the data set can be found.
2. Process metadata (i.e. where did the data originate from and what has been done to it) describes the original source of data, processing performed on the data set and the processing application.
3. Descriptive metadata (i.e. what does the data mean) consists of business and technical metadata. The business metadata describes the meaning of the data set from a business perspective (e.g. a link to the organizational policy to be used in evaluation of the data set) and its purpose for decision making (e.g. a link to the decision making policy to be used in evaluation of the data set). The technical metadata provides the technical information of the data set, such as a unique identifier, the language and size of the data, content description, data creator and creation place, content type and format, and required software to render and use the data.
4. Quality metadata (i.e. the applicability of the quality of data for its intended use) consists of the attributes (e.g. timeliness and accuracy) and the metrics that describe the quality of data.
5. Administrative metadata (i.e. how to access and use the data) describes the data provider, the applicable license(s) and access rights on the data set, the copyright holder and indicator of the data privacy level.

This work concentrates on quality metadata, assuming that other groups of metadata also exist.

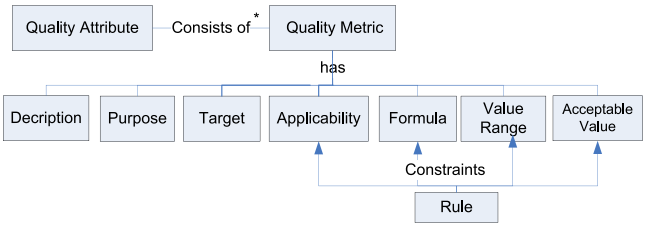

FIGURE 3. Properties of data quality metrics (adapted from [42]).

3) DESCRIPTION OF QUALITY METADATA

Table 1 describes the attributes of quality metadata. Each quality attribute describes a single aspect or construct of a quality. A quality attribute consists of one to several quality metrics that are measures of certain properties of the quality attribute (**Fig. 3**). Each metric has the following properties (adapted from [42]):

- Description: the description of the metric
- Purpose: the description of the metric purposes
- Target: where the metric can be used.
- Applicability: when the metric can be used.
- Formula: how the value for the metric is achieved.
- Value range: the range value for the metric measurement/evaluation.
- Acceptable value: the minimum measure accepted for the quality attribute.
- Rule: the set of constraints defining the set of targets of measurement, the set of value ranges for the measurement unit and the time when the metric is valid.

4) SELECTING QUALITY METADATA ATTRIBUTES FOR A DATA SET

The collected data can be of different types, such as a) any freely available data according to a company's interests,

TABLE 2. The application of social media data quality attributes.

Quality attribute	Applicability time	Metric (examples for twitter data)
Believability	Extraction, analysis	Evaluating the believability of a source (e.g. the identity of the authors): • Registration age = the time passed since the author registered his/her account, in days • Statuses_count = the number of tweets/comments/ questions at posting time • Followers_count = the number of people following this author • Friends_count = the number of people this author is following • Is verified = the author has a 'verified' account
Corroboration	Analysis	The amount of analyzed data sets from which the identified issue is recognized
Validity	Extraction, analysis	Estimation of the likelihood (0…1) of data validity in its purposed usage
Popularity	Extraction	• Popularity of source = the number of readers, followers, etc. • Popularity of information = the number of re-tweets, comments, questions, etc.
Relevancy	Extraction, processing, analysis	The amount of occurrence of relevant key words in title, subject and description
Timeliness	Extraction, processing, analysis	The data set creation date

originating from uncertain sources from the Internet, (e.g. data from web pages or from social media), b) deliberately collected external data from reliable or uncertain sources for certain internal process purposes, (e.g. for market analysis and competitor analysis), c) customer feedback data that can be reliable or uncertain, depending on the way the feedback is given, or d) a company's internal data, such as product data and production data. The collected data is classified according to data source types, such as social media data, feedback data, product data, competitor data, history data, or production data. This classification assists in selection of applicable quality attributes for the metadata of the given data set. The attributes are classified for each data source type. For example, the attributes applicable for social media data are described in Table 2. Thus, for example, each data set with the data source type ''social media data'' has quality metadata with the same quality attributes in a specific situation.

B. DATA QUALITY EVALUATION IN DATA PIPELINE

1) EVALUATION PHASES

In our approach, the metadata is managed in the following phases: data extraction, data processing, data analysis, and decision making. The first three phases follow the big data pipeline (Fig. 2). In the decision making phase, the analyzed data is visualized to the data user with varying views and varying users controls (Interfacing and visualization in Fig. 2); without user control, limited set of control functions or detailed visualization and control functions. The decision making based on the visualized data is the responsibility of the data user (according to decision making policies of the company).

Fig. 4 describes the different evaluation focuses and viewpoints on data. In data extraction, the focus is on the data source, when quality evaluation focuses on data provenance and the data quality from the viewpoint of the situation at hand, i.e., why the data was extracted. In data processing and analysis the focus is on data itself, evaluating the different quality aspects of the data. In decision making, the data is examined from the data user's viewpoint (i.e. data in context), when the evaluation focuses on data trustworthiness, i.e., how to ensure the trustworthiness of data in decision making. The data provenance and data quality assists in trustworthiness evaluation.

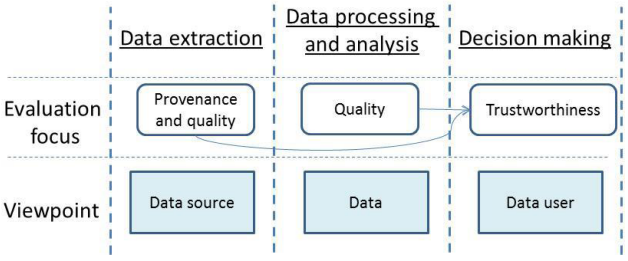

FIGURE 4. The focuses and viewpoints of data quality evaluation in the metadata management phases.

2) EVALUATION OF QUALITY ATTRIBUTES

The evaluation of quality attributes occurs in each metadata management phase. Quality evaluation can be qualitative or quantitative. Quantitative evaluation is a systematic and formal process, applicable in all metadata management phases. It relies on the existing knowledge of the company defined by the rules via a company's quality policies (see Section III C), and applies computational methods to achieve values for the metrics. The results of the quantitative evaluation are objective and more concrete than in the case of a qualitative evaluation. The quantitative evaluation can be automatized, and it can be performed by the company itself or it can be outsourced to third-party evaluation service providers.

Qualitative evaluation relies on the existing knowledge of the company, and also the experience and knowledge of the evaluator (expert or professional). The qualitative evaluation is applicable in data extraction, when the purpose of the data extraction is linked from business metadata to a company's

quality policies, and in decision making, when the value of the data is evaluated in the context of the current situation.

C. QUALITY METADATA MANAGEMENT IN BIG DATA SYSTEMS

To manage quality metadata, attributes and metrics, rules (see Fig. 3) are needed to define variability in quality, i.e. which quality attributes and metrics can be used and when. The rules can be described, for example, by a simple if-then-else structure or using some rule language, such as [43] and [44]. These rules should be part of a company's quality policy, which defines the principles and guidelines on how to manage quality in the company. The quality variability and quality policies are described in the next sub-sections.

1) DESCRIPTION OF QUALITY VARIABILITY

Different types of variation among quality attributes exist that describe a data set:

1. Target of attribute: Certain quality attributes are applicable only to certain data source types. For example, believability is an important attribute for data of which the origin is unclear. However, believability of a company's production data can be assumed to be high; thus the believability attribute is irrelevant. The quality attributes are selected based on the source type of the data set.
2. Applicability of attribute: Some of the quality attributes are applicable in the data extraction, some in the data processing or analysis, whereas some are applicable in all three phases. For example, corroboration cannot be evaluated for a single data set in data extraction, but it is important when evaluating several data sets in the analysis phase.
3. Target of metric: There are different metrics that can be used to evaluate a quality attribute. The selection of the metric is dependent on the data source type. For example, a different metric is used to evaluate corroboration in the case of twitter data or in the case of feedback data.
4. Applicability of metric: Different metrics can be used to evaluate the attribute in different phases. For example, the coverage of data is evaluated in data extraction phase by inspecting the amount and the content of data of the single data set, but in the analysis phase the coverage can be defined simply by the amount of the data sets

2) DATA QUALITY POLICIES

The data sets and metadata are administrated by the company's quality policies. The terms organizational policy and decision making policy have been adopted from [40]. The organizational policy defines the acceptable data sources, and describes all the elements from Fig. 3, such as the relevant quality attributes applicable to the context of the task at hand, the applicability time of the attributes, which evaluation metric should be used to evaluate each attribute, etc. Thus, the organizational policy consists of the set of rules that describe what and how to evaluate to achieve the data that can be trusted in a specific situation. A company can have several organizational policies, each of them applicable for different purposes of data collection.

The decision making policy describes which data sets are relevant for certain situations, how to weight quality attributes depending on the relevance of the different quality attributes for the task at hand, and how to perform the decision functions. The company can have several decision making policies, each of them describing the rules of how to make decisions in certain situations. Each policy can be applicable to different purposes of data collection/analysis or for different stakeholders. In addition, decisions are made during different stages of the product/service development process: in pre-development, development and post-development [5]. In the pre-development stage, the collected data is used in requirements specifications. During development, the data is used to identify modifications for the product/service and is an important input for further improvement. Finally, in the post-deployment stage, the data is used to optimize or innovate new features for a current or new product. The selection of the appropriate decision making policy is based on the existing experiences and knowledge of the company.

Both the organizational policy and decision making policy must be configurable by the data user to adapt the policies to the situation at hand. The user should be able, for example, to define the acceptable data sources, add new data sources, add new metrics/methods and configure the acceptable values for the quality metrics according to the context and purpose for the data collection. In the same way, during decision making, the user may want to configure acceptable values for the quality attributes for data set selection for decision making, or weighing quality attributes according to a new, changed situation.

D. PERFORMING THE DATA QUALITY EVALUATION AND MANAGEMENT IN BIG DATA SYSTEMS

This section describes how the previously introduced elements are used in the different evaluation phases in the big data pipeline, and what architectural elements are needed to enable data quality evaluation and management.

1) USING THE SOLUTION FOR QUALITY EVALUATION OF EXTRACTED DATA

Fig. 5 describes the activities that the end-user performs when using the solution for data extraction. These activities can be modified to be applicable also in the case of data processing and analysis. The main rationale for data collection is to assist a company in business and decision making; therefore, the meaning and purpose of the data collection must be defined beforehand (step 1 in Fig. 5). The purpose is later added into the descriptive business metadata (see section III A2 bullet 3). The metadata facilitates managing the data sets and enables the users to validate the value of data. The metadata is managed by the organizational and decision making policies,

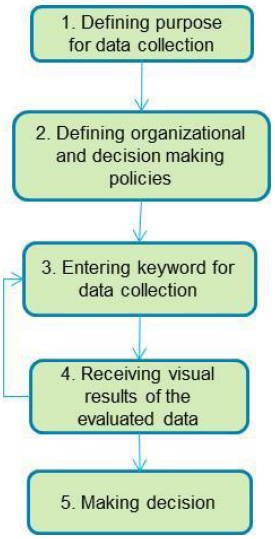

FIGURE 5. The user activities for data quality evaluation.

which must be defined applicable to certain business goals or certain types of purposes (Step 2 in Fig. 5). After that, the data and metadata management are automatically guided by the policies. Step 1 and Step 2 should be carefully defined, since they describe the reason and rules for data collection and evaluation. The end-user can collect data, for example, by entering a search keyword through the user interface (Step 3 in Fig. 5). The solution automatically extracts the data, evaluates the data quality, and finally visualizes it to the user according to the quality policies (Step 4). After seeing the results, the user may want to change metadata values (going back to the Step 3) and bring more data sets into the evaluation. Finally, the end-user makes business decisions based on the data (Step 5).

Our solution enables automatic data quality evaluation and management. Quantitative evaluation can be entirely automatized. Since the qualitative evaluation is managed mainly by human experts or professionals, it requires visualization of metadata to the user, and a user interface that enables the user to input values into the metadata (adding a new step between the steps 3 and 4).

2) CREATION OF QUALITY METADATA IN THE BIG DATA PIPELINE

Fig. 6 represents the data extraction, processing, analysis and decision making functionalities as an activity diagram. The functionalities are assisted by quality policies, in which the company's knowledge is presented by rules. In data extraction, the organizational policy facilitates the process by defining the acceptable data sources, and in selection of acceptable quality attributes, applicability time of the quality attribute and metrics and methods to evaluate the quality attributes. The applicable attributes are automatically provided when the data source type of the data set is known. The quality attributes are then evaluated using qualitative and/or qualitative evaluation. After extraction the imported data is stored in data storage. The quality metadata is created for the data set and the evaluated values for quality attributes are automatically inserted into the metadata. The metadata is stored in a metadata registry, separately from the data set.

In the same way, the organizational policy helps to select data sets for processing/analysis purposes. For example, the policy can set the value range for the quality attributes in metadata; only the data sets whose evaluated quality attributes fulfill the policy requirements defined for the processing/analysis phase are accepted, others are discarded. The organizational policy also assists in attaching the applicable quality attributes for the metadata of the data set and the metrics in this phase for evaluating the quality. After evaluation the quality metadata is created for the processed/analyzed data set and the evaluated values for quality attributes are inserted into the metadata.

During decision making, the decision making policy facilitates the selection of relevant data for the decision making purposes, e.g., by defining the important quality attributes and the minimum values for the selected data sets. That is, the policy defines which data sets are important for the situation at hand, and also validates their reliability and value for decision making. When evaluating the significance of a data set for a certain purpose, the decision making policy helps to weight the relevant quality attributes for the particular situation. The data is visualized to the data user with a visualization application with certain views and controls on data (defined in decision making policy). Decision making policy is always dependent on the company, its priorities and the goals and purposes for data gathering and analysis.

3) ARCHITECTURE FOR METADATA MANAGEMENT AND QUALITY MANAGEMENT

The architecture for the data, metadata management and quality management includes several elements of Fig. 2 with the following responsibilities:

- Extractor; extracts the data from data sources
- Temp data store; stores the extracted data temporarily
- Deep analytics; performs batch processing-based analysis for the collected data sets
- Analysis results store; stores the analysis results permanently
- Metadata management; responsible for creating, updating, storing and accessing the metadata
- Quality management; manages quality metadata for the data sets utilizing the company's quality policies and quality evaluator services. It includes the following complementary elements: Quality policy manager for the management of a company's quality policies, and Quality evaluator for evaluating the values for the metrics of the quality attributes.
- Metadata store; stores the metadata of the extracted, processed and analyzed data sets
- End-user application; provides the user interface to manage the data extraction, processing and analysis,

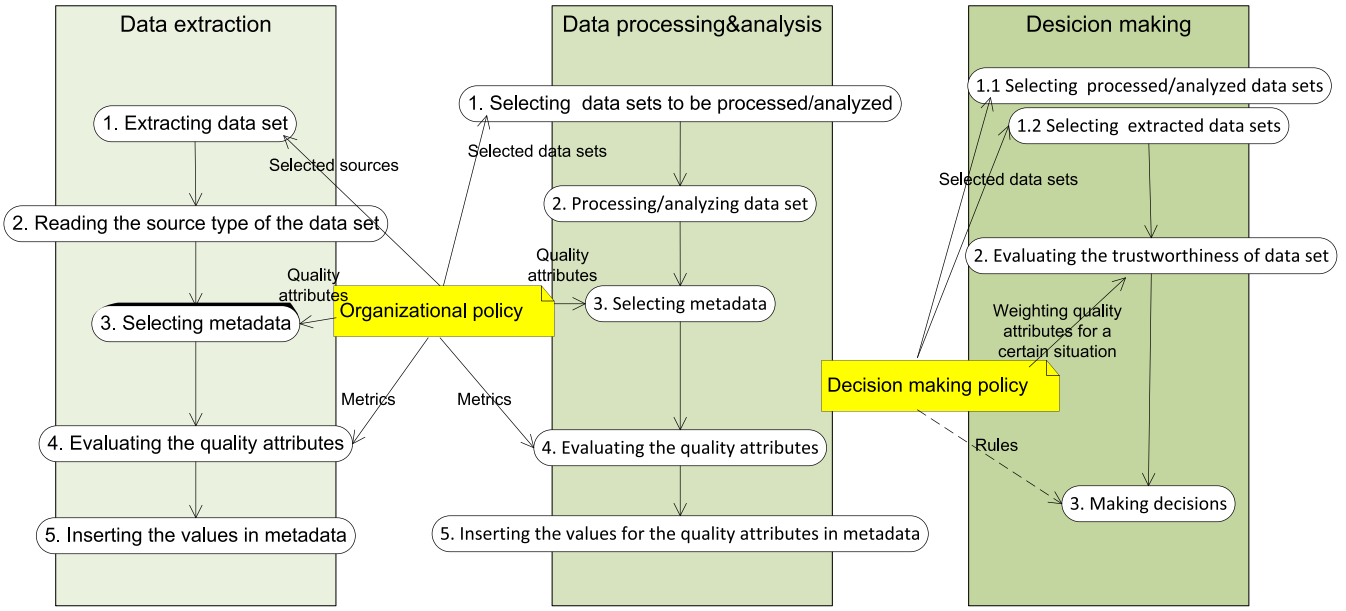

FIGURE 6. **Creation of quality metadata in different phases of metadata management in the data pipeline.**

enables the end-user to configure the quality policies, and visualizes the analysis results for the end-user.

Fig. 7 describes the architecture of the new element of Fig. 2; Metadata management, in more detail. Metadata management enables organization and management of the metadata of the data sets, and also the creation and management of quality metadata, enabling data quality evaluation. The architectural elements and their responsibilities are described in Table 3.

IV. CASE EXAMPLE

We demonstrate our solution using an industrial case example; the solution provides to the case company (a big data consulting company) insight regarding customer needs, which may facilitate R&D of the company. The data for the case example has been gathered by interviewing the case company's representatives. Also, the case example was implemented together with the case company, who wants to utilize social media data to find out what is discussed about their customers' products. The main purpose of the company is eventually to combine social media data with the company's own, internal data to achieve 'customer insight' that can be utilized in business decision making. The organizational and decision making policies have a great importance in quality evaluation; the definition of these policies is an organizational issue and is required as prerequisites for using the solution.

Fig. 8 describes an instantiation of elements in Fig. 2, illustrating the steps of data management in the case example at the architectural level.

Step 1 (Data Extraction and Analysis): At the data extraction phase, the end-user searches for relevant data using keywords. The keywords may be related, for example, to customers' products, and they are used for extraction of related tweets from Twitter. The tweets are extracted and saved into a temporary data store, and finally the sentiment of the tweets is analyzed. The case company has to define the acceptable data sources by the organizational policy before extraction of tweets. Step 1 of Fig. 8 is described in in more detailed in the following:

1.1 The end-user specifies keywords related to interesting commercial products.
1.2 DataExtractor extracts tweets via Twitter API based on the specified keywords (with a HTTP GET).
1.3 The tweets are saved into a temporary data store.
1.4 Deep analytics fetches the stored data sets from the TempData store after a certain time period.
1.5 Deep analytics performs sentiment analysis on the data sets. The aspect-based sentiment analysis [45] is used to analyze the sentiment of each individual aspect (words) in the discussion about the product, and to provide a sentiment score for the whole discussion.
1.6 The analysis results are saved into the analysis results store.

Step 2 (Metadata Creation and Data Quality Evaluation): This step focuses on creation of metadata in the big data pipeline. The metadata and related quality attributes are created based on the data sets of tweets (created in step 1) and the attributes are evaluated. The navigational, process, descriptive and administrative metadata are also created, but are not focused on in this paper. Step 2 of Fig. 8 is described in in more detailed in the following:

2.1 After saving the analysis results, Deep analytics notifies Metadata management to create metadata for

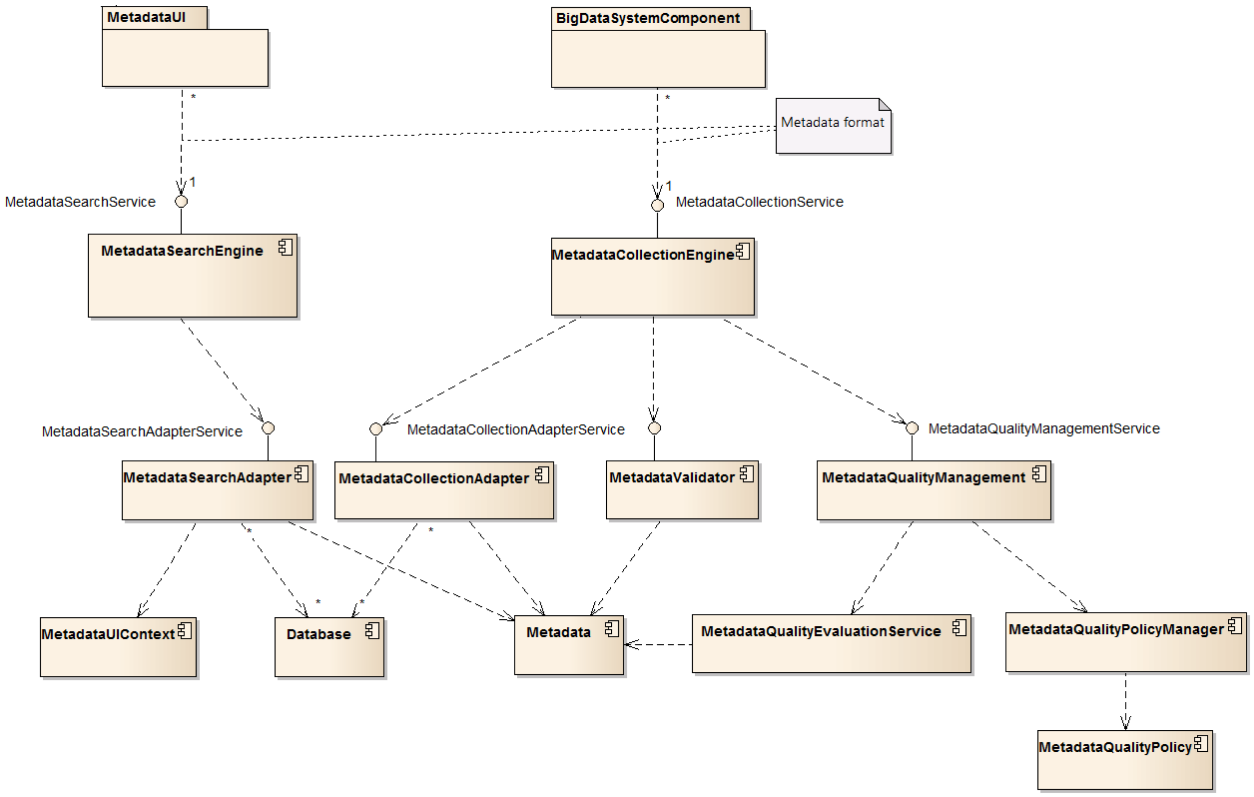

FIGURE 7. Structural view of metadata management architecture.

TABLE 3. Architectural elements of metadata management.

Element	Responsibility
MetadataCollectionEngine	Enables the extraction of metadata in a big data system via MetadataCollectionService API. The API enables external components of a big data system to input metadata to the Metadata store.
MetadataSearchEngine	Enables the searching for external components (e.g. through an UI) for metadata based on keywords and time via MetadataSearchService API.
Database	Stores the metadata of the extracted, processed and analyzed data sets (Metadata store implementation in Fig. 2).
MetadataSearchAdapter, MetadataCollectionAdapter	Adapters for translation of interaction between entities of a big data system and database.
Metadata	Contains definitions of quality, administrative, descriptive, process, and navigational metadata for reception via MetadataCollectionService API and storage to database.
MetadataUIContext	Contains metadata definitions (including additional quality attribute parameters) for publishing via MetadataSearchService API.
MetadataValidator	Validates the metadata received via MetadataCollectionService.
MetadataQualityManagement	Manages quality metadata for the data sets utilizing the company's quality policies.
MetadataQualityPolicyManager	Manages the company's quality policies, containing both the organizational and decision making policies.
MetadataQualityEvaluationService	Responsible for evaluating the values for the metrics of the quality attributes.
MetadataUI	Provides the user interface for visualization of metadata to the end-user.
BigDataSystemComponent	Component of a big data system, which provides metadata to metadata management.

the analyzed data set. In this step, provided information includes navigational, process, descriptive and administrative metadata.

2.2 Metadata management notifies Quality management to create appropriate quality metadata for the analyzed data set.

2.3 Quality management notifies the Quality policy manager to select the appropriate metadata quality attributes for the source type 'social media data' according to the quality policy.

2.4 The Quality policy manager returns the appropriate quality attributes for the analyzed data set

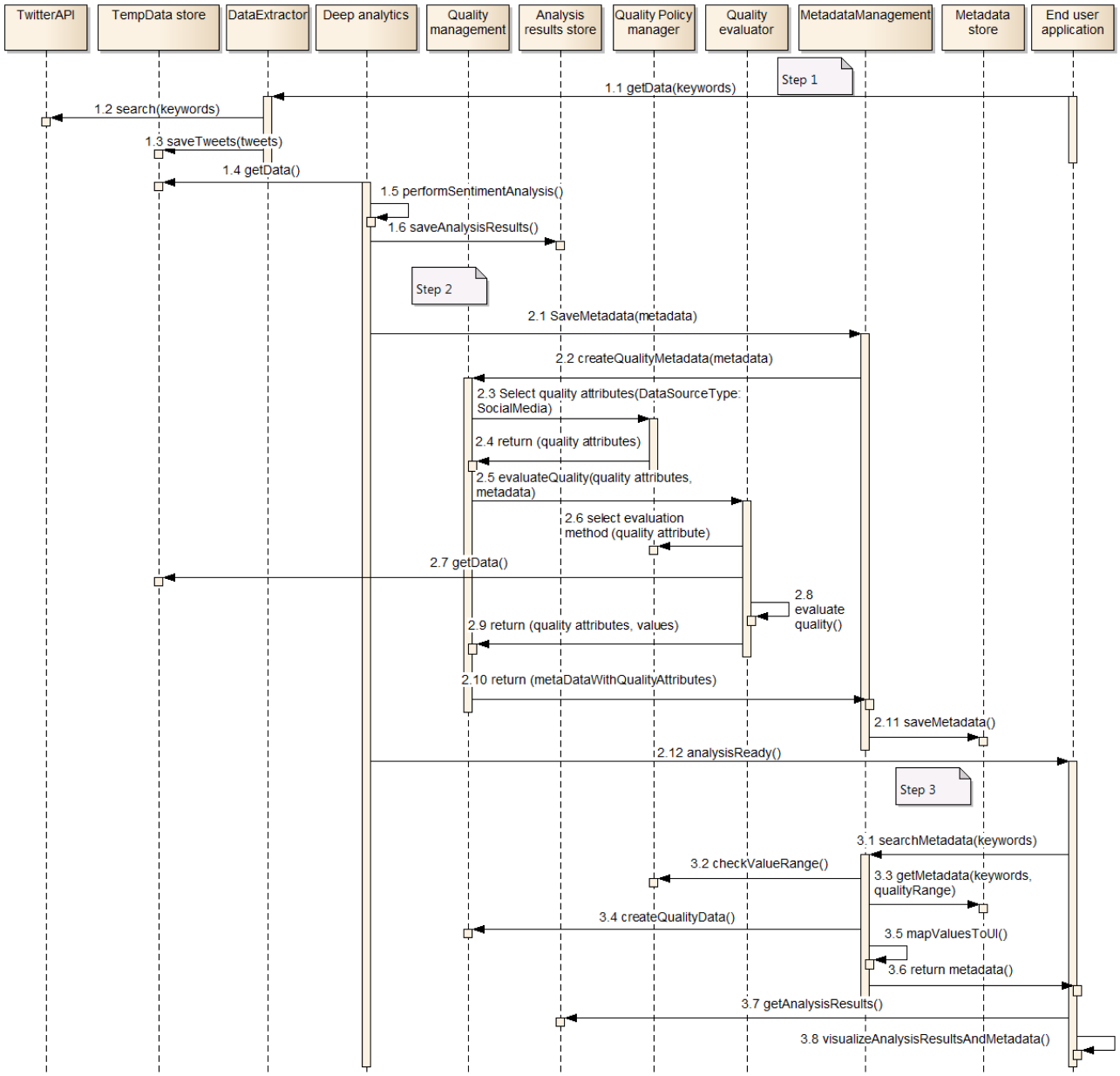

FIGURE 8. **Data and quality metadata management in big data architecture.**

(defined in organizational policy): timeliness and relevancy.

2.5 Quality management asks the Quality evaluator to evaluate the quality attributes.

2.6 For each quality attribute, the Quality evaluator checks for the appropriate metrics, evaluation methods and techniques defined in the organizational quality policy from the Quality policy manager.

2.7 The Quality evaluator fetches the data set (tweets) based on (navigational) metadata, which indicates location of the data set.

2.8 The Quality evaluator evaluates the following quality attributes: Timeliness is evaluated based on the timestamp of the metadata. Relevancy is determined based on the quality of the sentiment analysis algorithm (i.e. performance/quality of the analysis method), which is included into the metadata (in process metadata).

2.9 The Quality evaluator returns metadata with the quality attributes with values to Quality manager.

2.10 The Quality manager returns the values to Metadata management.

2.11 Metadata management writes the values into the quality metadata and saves the metadata into the Metadata store.

2.12 Deep analytics notifies the End-user application about a new analyzed data set.

Step 3 (Visualizing the Data to User for Decision Making): In this phase, the metadata is searched from the database for presentation to the end-user for decision making purposes. In this case example, the selected quality attributes include timeliness and relevancy. The relevant data is visualized to the end-user; the decision making policy defines the valuable data for the decision making by selecting only the data sets with the adequate quality attribute values. These policies are defined case-specific and applicable to the certain situation at hand. By changing the value range in a policy, the data sets with lower quality values can be selected. The following describes Step 3 of Fig. 8 in more detail:

3.1 The end-user (in this case; a decision maker) manages the data analysis through end-user application. An end-user wants to search interesting data sets with user-defined keywords (e.g. ''sentiment analysis'' and ''Product X''). The End-user application asks Metadata management to search the semantic keywords that are saved in the navigational metadata. The navigational metadata includes a list of semantic tags or keywords identifying the data set.

3.2 Metadata management checks the minimum values of the quality attributes of the data sets to be selected for the analysis (defined in decision making policy) from the Quality policy manager. For example, the selected data sets may not be older than one month, their relevancy must be at least 0.9 and believability must be at least 0.6. (value range $0 \ldots 1$).

3.3 Metadata management selects metadata sets, which include provided keywords and exceed the minimum values for the quality attributes from the Metadata store.

3.4 The timeliness attribute is recreated based on the current time.

3.5 Numerical values of quality attributes are mapped into human readable text for UI representation (e.g. timeliness value $> 0.7 \rightarrow$ 'very recent').

3.6 Metadata management returns the metadata to the end-user application.

3.7 The End-user application fetches the sentiment analysis data sets from the Analysis results store based on the (navigational) metadata.

3.8 Sentiment analysis results and metadata are visualized for the end-user. In the case example, the end-user prefers high relevancy of data prior to timeliness; thus the results are visualized in order of their relevancy.

As the data is visualized to the end-user, the end-user receives real time, validated information to support decision making. The company's decision makers then decide which actions to take. The company can have different levels of decision makers; the information is visualized according to the decision making policy. The decision making still requires a human and his/her expertise, and is assisted by the knowledge that the company has achieved (defined in the decision making policies).

In the case example, the data end-user receives the analysis results in order of their relevancy to the situation at hand. The user receives the positive and/or negative sentiment about the product, and uses this information, for example, to detect what kind of product features are desired and thus could be implemented and which features are negative and could be improved.

V. VALIDATION OF THE SOLUTION

The objective of the case example was to demonstrate the metadata and quality management with a social media use case. The implementation was conducted under DIGILE's Need for Speed (N4S) program[5] in collaboration with an industrial partner and VTT.[6] Metadata management was implemented and integrated with a big data use case as follows: The case company (company X) has built (into a public cloud) a system, which extracts tweets based on user-defined keywords, and performs sentiment analysis and visualization with a user interface (steps 1.1 - 1.6 in Fig. 8). We (VTT) provided the metadata management implementation, which is executed in VTT's separate, private cloud, and which provides a REST API for the big data system. The software implementation of the big data system was instrumented with calls to the metadata management interface (by company X) for transmitting of metadata information (step 2.1 in Fig. 8). VTT implemented the rest of the steps of Fig. 8 (from step 2.2. ahead), and also built a separate user interface into the private cloud for visualizing collected metadata for both organizations.

Currently, company X provides metadata information of extracted Twitter data sets, which is utilized as a basis for sentiment analysis. DataSourceType indicates the type of collected data sets, which can be utilized for determination of the relevancy attribute (step 2.8 in Fig. 8). Timeliness is determined based on the provided timestamp at the time of metadata extraction by comparison to the current time (step 3.4 in Fig. 8).

A. IMPLEMENTATION

When metadata management architecture was implemented, the technology choices, at least for metadata storage and API to the big data system, had to be determined (MetadataCollectionService and MetadataSearchService in Fig. 7). The technology choices are described and discussed in the following:

1) DATABASE FOR METADATA

Cassandra [46]. Metadata is saved into a column family, where a compound primary key for data was created based on a timestamp, and a parameter of descriptive metadata. An index had to be created into navigational metadata to

[5]http://www.n4s.fi/en/
[6]VTT Technical Research Centre of Finland.

enable searching based on keywords. Also, filtering has to be enabled in database queries based on keywords (with 'allow filtering'). This may lead to sub-optimal query latency, when compared to queries implemented with the primary key, which is very efficient in Cassandra [46]. Alternatively, a document oriented database (e.g. MongoDB) could be selected for storage of metadata due to the document structure of metadata.

2) METADATA API

XML over REST with Jersey. Alternatively, SOAP could have been selected as an implementation technology instead of REST. Earlier performance tests have indicated that REST has better performance than SOAP [47], [48]. The differences between REST and SOAP have been compared at the service level [48].

3) XML FORMAT VALIDATOR

Hibernate Validator. XML is an industry standard for platform-independent messaging. Alternatively, exchanged messages over REST could have been implemented with JSON. Differences between XML and JSON formats have been analyzed in terms of schema interoperability, serialization format, and message protocol [49].

B. VALIDATION

Initially, company X had an implementation of the social media use case. A requirement was to introduce only small changes to their existing software, which would enable extraction of metadata. Thus, VTT implemented metadata management architecture, which provided a REST interface for enabling straightforward instrumentation of software. One practical hindrance in the integration was the requirement for allowing cross-origin resource sharing [50]. This was caused by company X using a web browser within the enterprise domain, whereas data extraction and analysis was executed in the public cloud domain. In practice, the Access-Control-Allow-Origin header was needed in a HTTP response (to a HTTP OPTIONS request) for allowing access from the public cloud domain for extraction of metadata (a HTTP POST) with the web browser UI.

Metadata management implementation required about one month of development time, whereas instrumentation of a big data use case required one day of development time. No significant obstacles were discovered regarding the technological choices (see previous sub-section), when implementing extraction and search functionality of metadata. However, a more detailed analysis of performance and functionality may be needed, when the system is developed further with additional functionality.

The validation of the research solution was divided by company X focusing on big data use case R&D, while VTT designed and implemented metadata management architecture. Responsibilities were clearly divided in order to enable both organizations to focus on development of their software assets. REST API facilitated independent work on the activities by the organizations, and agreement of a common interface for integration. For the resource reasons, the existing demo of company X was used as a basis for implementation.

Currently, all steps of Fig. 8 have been implemented with the following limitations:

- Only one quality policy is implemented at this moment.
- The data in the case example was confidential data of the case company. This restricted the implementation of Step 2.7.
- Timeliness (time range) and keywords can be specified in the UI for searching of metadata (in step 3.1).

C. COMPARISON WITH RELATED WORKS

Only few works exists that relate to our solution. A quality evaluation framework for a big data pre-processing service is introduced in [51]. The framework is a generic solution that can be applied to different application domains, such as business, e-Health, IoT and social web. The quality evaluation pre-processing service is activated by a request with input data sources, output data destinations and a data quality profile. Each data input source has a data quality profile that contains reference to the actual data sources, output data and data quality rules. The pre-processing service includes the following architectural components: pre-processing activity selection, techniques selection, data quality selection, data profile optimization, data quality profile execution, quality control and data quality profile adapter. The quality evaluation service works iteratively by executing the defined processing activities and using the data profile adapter to change the data quality profile and notify the user about failed rules with suggestions on quality profile rules for better results. When compared to our solution, the main difference is the scope and focus. The scope of the proposed solution covers the latter part of the Data loading and pre-processing phase of our pipeline architecture introduced in Fig. 2. Our intent is to provide an architectural solution for managing quality of data in different phases of big data processing. Also, our solution focuses on using social media data in business decision making. Thus, all quality attributes of big data are not covered in our solution or in this quality framework.

Data quality centric big data architecture for federated sensor service clouds is introduced in [52]. The main contribution is the data quality (DQ)-aware virtualization of sensor services by enhancing each sensor feed's metadata with data quality attributes. The main components of the architecture are the DQ services catalog and DQ monitoring and adaptation component. Analysis is made in two phases: online feed analysis and batch analysis. The data quality model includes the following attributes: accuracy, error rate, availability, timeliness and validity. The main differences are in the architecture style and data quality model. This architecture focuses on connecting physical data sources to applications by applying a domain-specific data quality model. On the contrary, our solution focuses on big data processing and intends to manage the quality of unstructured social media data in each processing phase and applying quality policies

for adapting a quality model to the evaluation phase and data user's situation.

D. FUTURE DEVELOPMENT DIRECTIONS

The following development targets have been identified:

1. Implementation of several quality policies: Currently, each organizational policy is associated with the DataSourceType and one or more quality attributes. The QualityPolicyManager is responsible for initialization of the organizational quality policies. In the future more organizational quality policies could be defined for different social media types.
2. Evaluation of several quality attributes: Currently, our work is mainly an architecture for creation of quality aspects as part of overall metadata in a big data system (in the context of social media). Initially, the timeliness attribute provided a value based on the timestamp. In the future, algorithms will be developed, implemented, and validated for determination of several quality attributes in order to improve the utility of the solution.
3. Data/information search and user interface to quality management: The quality policies must be visualized to the user; the user must be able to, for example, update the quality policies, change the rules or add new acceptable data sources. The search based on other quality attributes must be implemented as well.

First of all, the different types of social media data (e.g. data from Twitter, Facebook or Instagram) should be able to be used together. Therefore, the definition of quality metrics for different types of social media data and rules for how to apply the properties of data quality metrics must be rationalized. Finally, the solution must be applied to different application domains and with different decision support systems to see how the quality attributes and rules are managed in different cases.

VI. CONCLUSIONS

This paper introduced a solution to evaluate the quality of data for business decision making purposes. The quality of data is evaluated in each data processing phase of the big data architecture with the help of quality metadata and quality policies. The solution may be adapted to different contexts, enabling the user to select the applicable quality attributes, evaluate them and apply them in a suitable way into a certain situation. The solution is also extendable; it allows inserting new data sources and data sets for data extraction, as well as new metrics and algorithms for data evaluation. The metadata enables location, retrieval and management of all the data sets, and the quality attributes and their values in metadata enable detection of the quality and value of data in a certain situation.

The solution was demonstrated with a case example where a company finds out the level of customer satisfaction regarding the quality of a product utilizing social media data. The solution was implemented with an industrial partner using a standard interface, which facilitated independent work of the company and the research organization, and functioned as a good communication tool for agreement with the integration. Several development targets were identified when demonstrating the solution. First of all, support for automating the quality attribute evaluation is required. The (semi-) automated adaptation of the organizational and decision making policies is required as well. However, the more knowledge the company achieves, the more the decision making process can be automatized with the help of quality policies.

At this moment the quality evaluation is limited to only a few quality attributes; the purpose is to extend the quality evaluation to include more quality attributes. One of the most important development targets is, however, to include other data source types, such as customer feedback data, product data and market analysis, to the quality evaluation to achieve 'customer insight' into business decision making.

REFERENCES

[1] S. R. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a Web of open data," in *The Semantic Web* (Lecture Notes in Computer Science), vol. 4825. Berlin, Germany: Springer-Verlag, 2007, pp. 722–735.

[2] A. Immonen, M. Palviainen, and E. Ovaska, "Requirements of an open data based business ecosystem," *IEEE Access*, vol. 2, pp. 88–103, Feb. 2014. DOI: 10.1109/ACCESS.2014.2302872

[3] S. Bhatia, J. Li, W. Peng, and T. Sun, "Monitoring and analyzing customer feedback through social media platforms for identifying and remedying customer problems," in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Mining (ASONAM)*, Aug. 2013, pp. 1147–1154.

[4] F. Antunes and J. P. Costa, "Integrating decision support and social networks," *Adv. Human-Comput. Interact.*, vol. 2012, Jan. 2012, Art. ID 9.

[5] A. Fabijan, H. H. Olsson, and J. Bosch, "Customer feedback and data collection techniques in software R&D: A literature review," in *Software Business* (Lecture Notes in Business Information Processing), vol. 210. Berlin, Germany: Springer-Verlag, 2015, pp. 139–153.

[6] R. Ferrando-Llopis, D. Lopez-Berzosa, and C. Mulligan, "Advancing value creation and value capture in data-intensive contexts," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2013, pp. 5–9.

[7] P. Pääkkönen and D. Pakkala, "Reference architecture and classification of technologies, products and services for big data systems," *Big Data Res.*, Jan. 2015. DOI: 10.1016/j.bdr.2015.01.001

[8] S. E. Madnick, R. Y. Wang, Y. W. Lee, and H. Zhu, "Overview and framework for data and information quality research," *J. Data Inf. Quality*, vol. 1, no. 1, 2009, Art. ID 2.

[9] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *J. Manage. Inf. Syst.*, vol. 12, no. 4, pp. 5–33, 1996.

[10] J. R. C. Nurse, S. S. Rahman, S. Creese, M. Goldsmith, and K. Lamberts, "Information quality and trustworthiness: A topical state-of-the-art review," in *Proc. Int. Conf. Comput. Appl. Netw. Secur. (ICCANS)*, Malé, Maldives, 2011, pp. 492–500.

[11] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of 'big data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, Jan. 2015.

[12] L. Cai and Y. Zhu, "The challenges of data quality and data quality assessment in the big data era," *Data Sci. J.*, vol. 14, no. 2, pp. 1–10, 2015.

[13] B. Ramesh, "Big data architecture," in *Studies in Big Data*, vol. 11, H. Mohanty *et al.*, Eds. India: Springer-Verlag, 2015. DOI: 10.1007/978-81-322-2494-5_2

[14] M. Chen *et al.*, "Data, information, and knowledge in visualization," *IEEE Comput. Graph. Appl.*, vol. 29, no. 1, pp. 12–19, Jan./Feb. 2009.

[15] *Software Engineering—Product Quality. Part 1: Quality Model*, ISO/IEC Standard 9126-1, ISO/IEC, 2001, p. 25.

[16] National Information Standards Organization, *Understanding Metadata*. Bethesda, MD, USA: NISO Press, 2004.

[17] W3C. (2007). *Web Services Policy 1.5—Framework (W3C Recommendation)*. [Online]. Available: http://www.w3.org/TR/ws-policy/
[18] *Quality Management Systems—Requirements*, ISO Standard 9001:2008, ISO, 2008.
[19] I. Gorton and J. Klein, "Distribution, data, deployment: Software architecture convergence in big data systems," *IEEE Softw.*, vol. 32, no. 3, pp. 78–85, May/Jun. 2015.
[20] N. Foshay, A. Mukherjee, and A. Taylor, "Does data warehouse end-user metadata add value?" *Commun. ACM*, vol. 50, no. 11, pp. 70–77, 2007.
[21] B. E. Bargmeyer and D. W. Gillman, "Metadata standards and metadata registries: An overview," in *Proc. Int. Conf. Establishment Surv. II*, Buffalo, NY, USA, 2000, pp. 1–10.
[22] *Common Warehouse Metamodel (CWM) Specification*, OMG document ad/99-09-01, OMG, 1999. [Online]. Available: http://www.omg.org
[23] *Information Technology—Metadata Registries (MDR)—Part 3: Registry Metamodel and Basic Attributes*, ISO/IEC Standard 11179-3:2003(E), International Organization for Standardization, Geneva, Switzerland, 2003.
[24] National Information Standards Organization, *The Dublin Core Metadata Element Set, Version 1.1*. Bethesda, MD, USA: NISO Press, 2012. [Online]. Available: http://www.dublincore.org/documents/dces
[25] *Software Engineering—Product Quality. Part 2: External Metrics*, ISO/IEC Standard TR 9126-2:2003, 2003.
[26] *Software Engineering—Product Quality. Part 3: Internal Metrics*, ISO/IEC Standard TR 9126-3:2003, 2003.
[27] A. Immonen and E. Niemelä, "Survey of reliability and availability prediction methods from the viewpoint of software architecture," *Softw. Syst. Model.*, vol. 7, no. 1, pp. 49–65, 2008.
[28] E. Ovaska, A. Evesti, K. Henttonen, M. Palviainen, and P. Aho, "Knowledge based quality-driven architecture design and evaluation," *Inf. Softw. Technol.*, vol. 52, no. 6, pp. 577–601, 2010.
[29] E. Niemelä and A. Immonen, "Capturing quality requirements of product family architecture," *Inf. Softw. Technol.*, vol. 49, nos. 11–12, pp. 1107–1120, 2007.
[30] R. Kazman, M. Klein, and P. Clements, "ATAM: Method for architecture evaluation," Carnegie Mellon Univ., Softw. Eng. Inst., Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2000-TR-004, Aug. 2000. [Online]. Available: http://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13706.pdf
[31] L. Dobrica and E. Niemelä, "A survey on software architecture analysis methods," *IEEE Trans. Softw. Eng.*, vol. 28, no. 7, pp. 638–653, Jul. 2002.
[32] Y. Gil and D. Artz, "Towards content trust of Web resources," *Web Semantics, Sci., Services, Agents World Wide Web*, vol. 5, no. 4, pp. 227–239, 2007.
[33] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, "An approach to evaluate data trustworthiness based on data provenance," in *Proc. 5th VLDB Workshop Secure Data Manage.*, vol. 5159. 2008, pp. 82–98.
[34] F. Naumann and C. Rolker, "Assessment methods for information quality criteria," in *Proc. 5th Int. Conf. Inf. Quality*, Boston, MA, USA, 2000, pp. 148–162.
[35] J. R. C. Nurse, I. Agrafiotis, S. Creese, M. Goldsmith, and K. Lamberts, "Building confidence in information-trustworthiness metrics for decision support," in *Proc. 12th IEEE Int. Conf. Trust, Secur., Privacy Comput. Commun. (TrustCom)*, Melbourne, VIC, Australia, Jul. 2013, pp. 535–543.
[36] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 675–684.
[37] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *Proc. Int. Conf. Web Search Data Mining (WSDM)*, 2008, pp. 183–194.
[38] C. Bizer, "Quality-driven information filtering in the context of Web-based information systems," Ph.D. dissertation, Dept. Econom. Sci., Freie Univ. Berlin, Berlin, Germany, 2007.
[39] C. Bizer and R. Cyganiak, "Quality-driven information filtering using the WIQA policy framework," *Web Semantics, Sci., Services, Agents World Wide Web*, vol. 7, no. 1, pp. 1–10, 2009.
[40] S. S. Rahman, S. Creese, and M. Goldsmith, "Accepting information with a pinch of salt: Handling untrusted information sources," in *Security and Trust Management* (Lecture Notes in Computer Science), vol. 7170. Berlin, Germany: Springer-Verlag, 2011, pp. 223–238.
[41] E. Bertino and H.-S. Lim, "Assuring data trustworthiness—Concepts and research challenges," in *Secure Data Management* (Lecture Notes in Computer Science), vol. 6358. Berlin, Germany: Springer-Verlag, 2010, pp. 1–12.
[42] E. Niemelä, A. Evesti, and P. Savolainen, "Modeling quality attribute variability," in *Proc. 3rd Int. Conf. Eval. Novel Approaches Softw. Eng.*, Funchal, Portugal, 2008, pp. 169–176.
[43] V. Luukkala and I. Niemelä, "Enhancing a smart space with answer set programming," in *Semantic Web Rules*, M. Dean, J. Hall, A. Rotolo, and S. Tabet, Eds. Berlin, Germany: Springer-Verlag, 2010, pp. 89–103.
[44] W3C. (2012). *SPARQL Query Language for RDF*. W3C Recommendation. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query
[45] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, 2013.
[46] DataStax. *CQL for Cassandra 2.2*. [Online]. Available: http://docs.datastax.com/en/cql/3.3/cql/cqlIntro.html, accessed Aug. 10, 2015.
[47] T. Aihkisalo and T. Paaso, "Latencies of service invocation and processing of the REST and SOAP Web service interfaces," in *Proc. IEEE 8th World Congr. Services*, Jun. 2012, pp. 100–107.
[48] G. Mulligan and D. Gracanin, "A comparison of SOAP and REST implementations of a service based interaction independence middleware framework," in *Proc. Winter Simulation Conf.*, Austin, TX, USA, Dec. 2009, pp. 1423–1431.
[49] J. Delgado, "Service interoperability in the Internet of Things," in *Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence* (Studies in Computational Intelligence), vol. 460. Berlin, Germany: Springer-Verlag, 2013, pp. 51–87.
[50] A. van Kesteren. (2014). *Cross-Origin Resource Sharing*. W3C Recommendation. [Online]. Available: http://Www.W3.Org/TR/Access-Control/
[51] I. Taleb, R. Dssouli, and M. A. Serhani, "Big data pre-processing: A quality framework," in *Proc. IEEE Int. Congr. Big Data*, New York, NY, USA, Jun./Jul. 2015, pp. 191–198.
[52] L. Ramaswamy, V. Lawson, and S. V. Gogineni, "Towards a quality-centric big data architecture for federated sensor services," in *Proc. IEEE Int. Congr. Big Data*, Santa Clara, CA, USA, Jun./Jul. 2013, pp. 86–93.

ANNE IMMONEN received the M.Sc. degree in information processing science from the University of Oulu, Finland, in 2002. She is currently a Research Scientist with the VTT Technical Research Centre of Finland. Her main research interests include reliability in service engineering, in particular, in the context of digital service ecosystems. Her current research interests include the data and service ecosystems, big data, and the quality and trustworthiness of data.

PEKKA PÄÄKKÖNEN received the M.Sc. degree in information technology from the University of Oulu, Finland, in 2002. He is currently a Senior Research Scientist with the VTT Technical Research Centre of Finland. His research interests include distributed computing, big data technologies, databases, and software performance.

EILA OVASKA received the Ph.D. degree from the University of Oulu, in 2000. Prior to 2000, she was a Software Engineer, a Senior Research Scientist, and the Leader with the Software Architectures Group, VTT Technical Research Centre of Finland. Since 2001, she has been a Research Professor with the VTT Technical Research Centre of Finland and an Adjunct Professor with the University of Oulu. She has co-authored over 150 scientific publications. Her current areas of interest are service architectures, self-management systems, and knowledge oriented service engineering. She has acted as a Workshop and Conference Organizer and Reviewer for scientific journals and conferences.

Publication V

Towards certified open data in digital service ecosystems

Software Quality Journal,
published online 21 June 2017. 41 p. In press.

Software Qual J
DOI 10.1007/s11219-017-9378-2

Towards certified open data in digital service ecosystems

Anne Immonen[1] · **Eila Ovaska**[1] · **Tuomas Paaso**[1]

Abstract The opportunities of open data have been recently recognized among companies in different domains. Digital service providers have increasingly been interested in the possibilities of innovating new ideas and services around open data. Digital service ecosystems provide several advantages for service developers, enabling the service co-innovation and co-creation among ecosystem members utilizing and sharing common assets and knowledge. The utilization of open data in digital services requires new innovation practices, service development models, and a collaboration environment. These can be provided by the ecosystem. However, since open data can be almost anything and originate from different kinds of data sources, the quality of data becomes the key issue. The new challenge for service providers is how to guarantee the quality of open data. In the ecosystems, uncertain data quality poses major challenges. The main contribution of this paper is the concept of the Evolvable Open Data based digital service Ecosystem (EODE), which defines the kinds of knowledge and services that are required for validating open data in digital service ecosystems. Thus, the EODE provides business potential for open data and digital service providers, as well as other actors around open data. The ecosystem capability model, knowledge management models, and the taxonomy of services to support the open data quality certification are described. Data quality certification confirms that the open data is trustworthy and its quality is good enough to be accepted for the usage of the ecosystem's services. The five-phase open data quality certification process, according to which open data is brought to the ecosystem and certified for the usage of the digital service ecosystem members using the knowledge models and support services of the ecosystem, is also described. The initial experiences of the

✉ Anne Immonen
anne.immonen@vtt.fi

Eila Ovaska
eila.ovaska@vtt.fi

Tuomas Paaso
tuomas.paaso@vtt.fi

[1] VTT Technical Research Centre of Finland, Digital Systems and Services, P.O. Box 1100, FI-90571 Oulu, Finland

Published online: 21 June 2017

still ongoing validation steps are summarized, and the concept limitations and future development targets are identified.

Keywords Quality of data · Quality policy · Digital service ecosystem · Semantics · Interoperability · Knowledge sharing

1 Introduction

Digital service providers have been increasingly interested in digital service ecosystems as the ecosystem-based service development provides several advantages, including collaborative innovation and value co-creation among ecosystem members. In a digital service ecosystem, the ecosystem members can utilize and share common assets and knowledge, nevertheless act independently. The product of a digital ecosystem, a digital service, can be anything that is intended to be entirely automated and can be delivered digitally through an information infrastructure. Recently, freely available open data has increasingly interested service providers, as this data has been identified to provide several business benefits, such as new data-based content, ideas and basic functions, increased understanding about business opportunities, improved competitiveness, and potential new customers (Immonen et al. 2014). Especially open social media data interests companies as it can provide insight into consumers' opinions, preferences, and requirements considering the company or its products/services (Bhatia et al. 2013; Antunes and Costa 2012; Fabijan et al. 2015), thus enabling the companies to achieve "customer insight" into business decision-making (Immonen et al. 2015a). Bringing open data into the context of ecosystem-based service engineering delivers all these benefits available to ecosystem members and also facilitates the utilization of open data in digital service engineering.

Open data is based on the idea that certain data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents, or other mechanisms of control (Auer et al. 2007). The open data concept has evolved over the 10 years since its first definitions. The tendency in many countries has been to open the administrative data (Poikola et al. 2011), and several local and global open data portals already exist that help people to create and share data and knowledge. Open data typically originates from enormous amounts of different kinds of sources, and it can be structured (with a strict data model), semi-structured (with an evolving data model), or unstructured (not associated with any data model). The utilization of this kind of data requires knowledge about its provenance, quality, and trustworthiness to ensure that the data is what it is expected to be. Data quality can be defined as data that is fit for use by data consumers (Wang and Strong 1996). The evaluation of data quality is challenging due to the facts that there are no agreed definitions of quality attributes, and the data quality cannot be judged without considering the context at hand (Nurse et al. 2011). The growing amount of semi- and unstructured data, new ways of delivering information, and users' changed expectations and perceptions of data quality (Madnick et al. 2009) further provide new challenges in data quality evaluation. At the same time, this dictates that new quality evaluation means and methods are required to verify the quality of open data. The importance of quality evaluation is emphasized in a digital service ecosystem, where poor quality of data affects several digital services and, in that way, the whole trustworthiness of the ecosystem.

Therefore, in the digital service ecosystem, data quality evaluation should be one of the key activities supported by the ecosystem's assets. The main question is how the ecosystem can guarantee the quality of open data.

According to the survey on the state of the practice among industry (Immonen et al. 2014; Immonen et al. 2013), quality assurance of open data is the biggest obstacle for its exploitation in digital service development. The contribution of this research is to specify the concept of an open data based digital service ecosystem (called the EODE concept), in which the ecosystem ensures the quality of open data utilized in digital services. In this approach, open data is provided as a service for the ecosystem's usage. The purpose of the concept is to verify the trustworthiness and the quality of open data, thus, to ensure that the data comes from reliable sources, and its quality is good enough to be accepted for the usage of the ecosystem's services. The members of the ecosystem do not have to be familiar with the metrics or techniques for data quality evaluation, but the ecosystem is responsible for certifying the quality of data that can be then utilized by the ecosystem members. The EODE concept includes the ecosystem's capability model with activities for the quality evaluation of the open data source, open data itself, and open data services, and it provides knowledge management models and ecosystem support services to enable these activities. The EODE supports the businesses of both the open data providers and the digital service providers. The open data providers reach more users (and thus more income) for the data when they pay more attention to data quality; data with poor quality is not selected for the ecosystem. The service providers receive more satisfied consumers when they provide trustworthy data via digital services. The ecosystem also provides other benefits to its members, such as finding partners and customers, and ways to deliver services and data. In addition, the EODE provides the possibilities and business potential for other support service providers as well, such as for analysis and monitoring service providers.

The EODE concept includes a quality certification process for open data, which specifies how the knowledge and support services of the ecosystem are utilized to carry out the quality evaluation. Thus, the process is a kind of instantiation and a guideline of the knowledge and the support services necessary to implement the quality certification of open data. Data certification contains several aspects, such as legal, practical, technical, and social aspects. Thus, besides data quality, data privacy, availability, and licensing aspects must also be considered when making decisions to accept the data for usage. However, data quality, i.e., the ability of the data to be fit for use by data consumers (Wang and Strong 1996), is the first aspect that must be ensured. If the quality is not good enough, there is no need to evaluate the other aspects. Therefore, this research concentrates purely on the technical quality aspects of open data. The quality certification process enables bringing open data to the ecosystem, transforming it to a usable form for the ecosystem, validating it against its intended usage, monitoring the data sources and the usage of the data, and continuously evaluating the quantified value of the open data service, thus, certifying the quality of the data for the ecosystem and its members.

This paper is organized according to the following: Section 2 presents the background for this research; the basic terminology is first defined, after which our earlier research on open data based business ecosystems, the quality evaluation of open data, and service engineering in ecosystems are presented. These are used as the basic and starting point for this research and are combined and refined to form a full open data based service

ecosystem concept. Studies related to this research are presented to understand the shortage to which this paper tends to respond, including concepts of the open data ecosystem (and the current status and development of open data), the quality evaluation of open data, and ecosystem-based digital service engineering. Finally, Section 3 introduces the concept of the EODE; Evolvable Open Data based digital service Ecosystem. The EODE is represented from two viewpoints: the ecosystem and the service providers. Section 4 introduces how the elements of EODE are utilized to implement the open data quality certification process. The process consists of five validation phases with related activities, required support services, knowledge assets and related evaluation targets, quality attributes, and metrics. Section 5 presents the analyses and discussion, consisting of the current validation of the concepts of the EODE, and limitations, open issues and future research targets. Finally, Section 6 presents conclusions drawn.

2 Background and related works

2.1 Terminology

The following terminology is used in this paper:

Data—Data that is produced by observing, monitoring, or using questionnaires, but has not yet been processed for any specific purpose.
Open data—Data that it is freely available to everyone to use and republish as they wish, without restrictions of copyrights, patents, or other mechanisms of control (Auer et al. 2007).
Information—Data that is refined and processed for assigning meaning to the data (Chen et al. 2009).
Quality of data—Data that is fit for use by data consumers (Wang and Strong 1996).
Data quality certification—Confirmation that the open data is trustworthy, and its quality has been verified according to strict quality policies.
Metadata—A standardized way to describe the semantics of data.
Policy—A collection of alternative tasks and rules, each of which represent a requirement, capability, or other property of behavior (W3C 2007).
Ecosystem policy—Description of the principles, strategies, tactics, and guidelines of the ecosystem that are common to all ecosystem members.
Organizational data policy—Description of the principles and guidelines required to effectively manage and exploit the data/information resources of a company.
Open data ecosystem—A free-formed community of organizations each of which have their own part and know-how in the data-based business.
Open data service—A service that encapsulates the open data, providing the open data as a service.
Digital service—A service that utilizes the open data, is entirely automated, and can be anything that can be delivered through an information infrastructure, e.g., web, mobile devices, or any other forms of delivery.
Digital service ecosystem—An open, loosely coupled, domain-clustered, demand-driven, self-organizing environment, in which digital services are created in value networks under the common ecosystem regulation.

2.2 Our earlier studies as a starting point for the research

The earlier studies by the authors are used as the basis and starting point for this research and are therefore presented in the following sub-sections.

2.2.1 Motives for the research

While examining the usage of open data in Finland, industry interviews were performed in 2013 pertaining to open data in business (Immonen et al. 2014; Immonen et al. 2013). It was discovered that there exists huge interest in open data and its exploitation in business. However, serious barriers were found to exist that prevent the fluent utilization of open data. These concern the lack of a standard description of data sources and APIs, as well as the uniform format for the data. Furthermore, the management of data privacy and varying licensing conditions and data quality were seen as highly important issues but have not been solved yet. However, the low quality of data and changes in data quality were seen as risks that complicate or even prevent the open data utilization in business (Immonen et al. 2014). In Immonen et al. (2014), the data broker actor is defined to include the role of data promoters that maintain "a list" of available data in the ecosystem and the quality of data, price, applied licenses, etc. Since the quality of the data was detected to be unknown, a need was identified for a data quality verification service in the ecosystem.

Now, two and a half years after the first interviews, new interviews were performed among the same industry representatives. It was detected that although data quality was seen as highly important, no significant progress had occurred in 2 years. The companies conceded that interest in open data and its exploitation in business still exists, and they have also recognized that the demand for open data from authorities, companies, and individuals has increased. Moreover, opening of data is also done in a smaller scope: contract-based exchange of data between companies is seen as a working collaboration model. The same main challenges remains: (1) the data that the companies are interested in is not available, (2) the data is not free of charge, and (3) there is uncertainty about the quality of the available data. Thus, the problems are related both to business and to used technology and raise the following questions: (i) What business reasons are there to produce open data and how can it be marketed? (ii) How can the use of open data be made profitable in service development? In summary, the use of open data has slowly progressed, nevertheless, several obstacles remain that must be removed. And, quality assurance of open data is the greatest obstacle to its exploitation in digital service development.

2.2.2 Ecosystem actors

In Immonen et al. (2014), the actors and their roles in the open data ecosystem from the business viewpoint are defined. These actors include the following: (1) data providers make data available to other stakeholders; (2) data brokers promote the data in the ecosystem, distribute it through the communication channels, and match the demanded and provided data; (3) service providers produce supporting services related to the data to be utilized in applications; (4) Application developers use the available data and services and develop applications for the data; (5) Application users are the data end-users that consume the data and services with the help of applications; and (6) infrastructure and tool providers provide utility services

to all the actors so they are able to act in the ecosystem. Furthermore, in this research on digital service ecosystem (Immonen et al. 2015b) defined the actors of the digital service ecosystem are defined from the service engineering viewpoint. These include the following: service providers that provide digital services to be used by other ecosystem members or consumers, service brokers that promote and deliver the services and match the demand with the best available services, service consumers that are the actual users of the services, and infrastructure providers that provide the utilities for acting in the ecosystem.

Open data-based digital service ecosystem merges and refines the actors both from the open data ecosystem and service ecosystem (see Fig. 1). The roles of infrastructure and tool providers remain the same. Open data service providers encapsulate the open data and provide the data as utility services, thus enabling the utilization of the open data in digital services. The data owner has data sovereignty and, thus, specifies the terms and conditions of use of the data (Boris et al. 2016). Digital service providers provide digital services that utilize the open data. Ecosystem support service providers provide services that support extracting, monitoring, and evaluating the data and, thus, assist in managing open data and its quality in the ecosystem. Finally, the digital service consumers utilize the data with the help of digital services.

2.2.3 Ecosystem capability and infrastructure

In Immonen et al. (2015b), the elements of the digital service ecosystem that influence service engineering in the ecosystem (see Fig. 2) are defined. The main elements, ecosystem members, infrastructure, capabilities, and digital services, are classified according to (Ruokolainen 2013). The capability of the ecosystem defines the properties of the ecosystem and how these are implemented using the infrastructure services (Immonen et al. 2015b). Thus, the capabilities define the purpose of the ecosystem, its ability to perform actions, and the rules for how to operate in the ecosystem. The actions and rules address the following:

1. Governance and regulation actions of the ecosystem (Immonen et al. 2015b) for

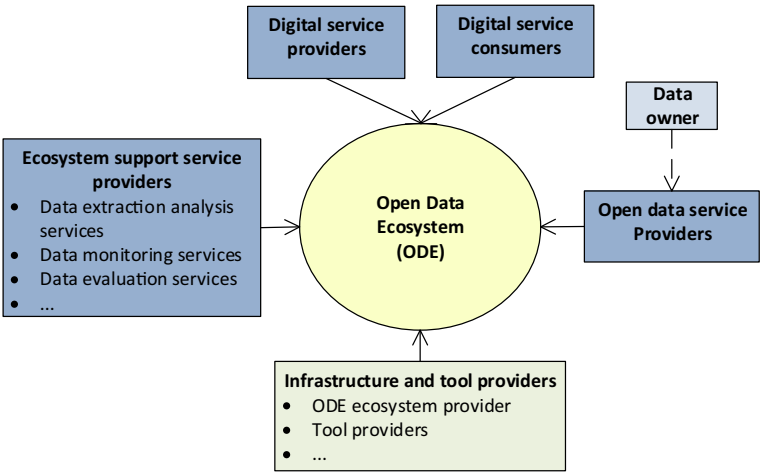

Fig. 1 The actor roles in an open data-based digital service ecosystem

- Directing, monitoring, and managing the ecosystem: these include, for example, rules of trusted collaboration establishment, interactions rules, and how to join and leave the ecosystem
- Directing and managing service engineering: these include, for example, rules for describing and delivering services and managing knowledge.

2. Service engineering-related actions

- Provide reusable assets for defining requirements (both functional and quality)
- Assist in the matchmaking of services
- Provide reusable assets for quality requirements specification, quality modeling, and quality evaluation of digital services

The infrastructure of the digital service ecosystem provides the knowledge models and services for implementing the ecosystem's capabilities. These include the following (Immonen et al. 2015b):

- A domain model: describes the concepts of the domain, their relations with each other, e.g., domain-specific quality attributes, rules, and policies
- A knowledge management model: describes the knowledge, know-how, and assets of the ecosystem
- A service engineering model: describes how the services are co-innovated and co-engineered in the ecosystem
- Ecosystem support services: implement the actions of the capability model

The elements described in (Immonen et al. 2015b) do not consider open data and the quality of data in digital service engineering. In this study, the capability model has been refined to include the open data related actions. The focus will be on ecosystem

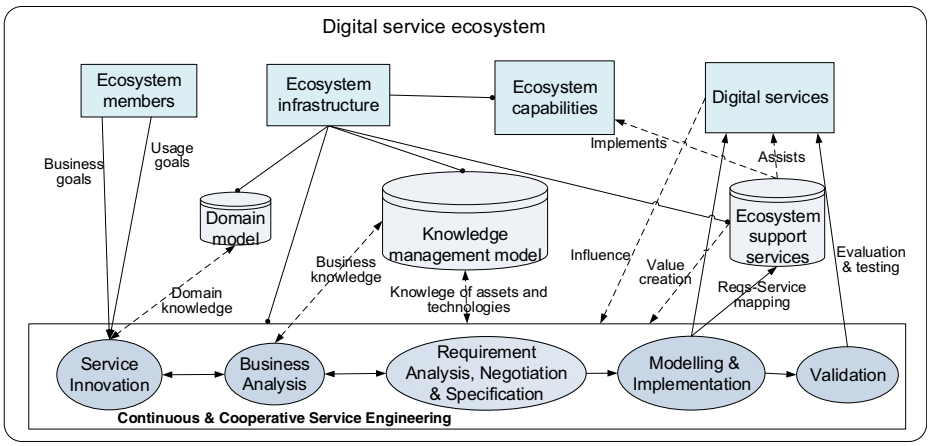

Fig. 2 Continuous cooperative service engineering in a digital service ecosystem (simplified from (Immonen et al. 2015b))

infrastructure and capabilities, including a domain model, a knowledge management model, and ecosystem support services. The content of these elements have been refined to include the activities, models, and services necessary to certify the open data and enable data utilization in service engineering in the ecosystem. Certification in this research means purely data quality certification; the other issues that concern assessing the extent of the open data (HM Government Cabinet Office 2012), such as access, licenses, and privacy, are beyond the scope of the present research.

2.2.4 Quality evaluation of open data

In Immonen et al. (2015a), the elements and phases of open data (social media data) quality evaluation in big data architecture are defined. The data is evaluated in data extraction, processing, and analysis phases with the help of organizational policies, and, finally, its value in decision-making is evaluated using a decision-making policy. The data is managed with the help of metadata, which is again managed utilizing the metadata management component/element included in the big data architecture.

The paper follows the five-star scheme (HM Government Cabinet Office 2012) and goes beyond that by defining "open data services" that make the data available for different service developers that want to utilize the data. Earlier work has focused on data quality and quality evaluation inside one company; it has not considered the ecosystem context. In this work, the purpose is to present how to validate the quality of data in the context of the digital service ecosystem. Therefore, the term "organizational policy" is not used, but it is distributed into the data filtering policy and evaluation policy that manage the data quality certification in the ecosystem. The aim is to obey the earlier definitions of open data (Fig. 3) and go further by providing the support services that are the first movement towards automated quality certification of open data in the context of digital service ecosystem (the green "future" box in Fig. 3).

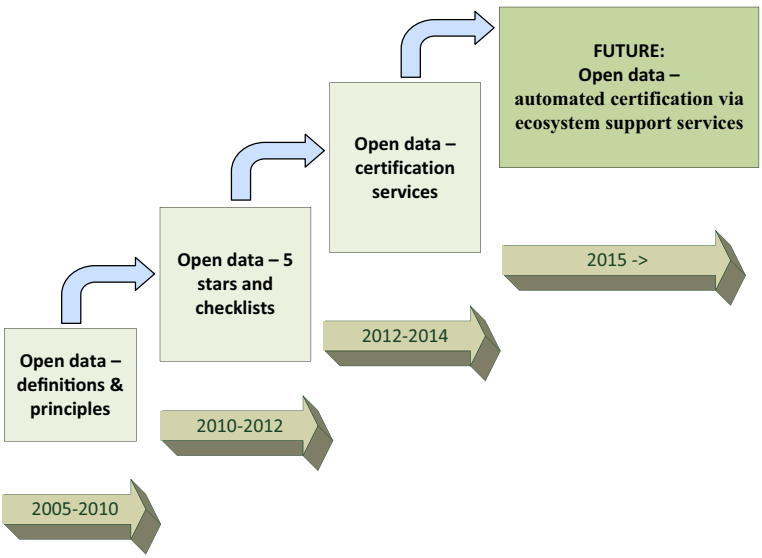

Fig. 3 Open data development stairs

2.3 Related research

2.3.1 Concepts of open data ecosystems

The concept of "open data" most notably has its roots in Great Britain, which has advanced the Open Government Data ecosystem over the past 15 years. The major breakthrough in the era of open data was in 2009 when both Great Britain and the USA launched their first data portals. Since then, the tendency in many countries has been to obligate to open the data of the public sector collected along with tax revenues. There exist many foundations and initiatives that "push" organizations to open their data, such as the Open Knowledge Foundation,[1] the Open Data Institute (ODI),[2] the Global Open Data Initiative,[3] and the INSPIRE[4] directive of the European Union. However, the "pull" mode has received less attention. Therefore, the data holders do not know the demand for the data that they own, or the possibilities that their data would provide to some other stakeholders. Some attempts already exist that tend to untangle the demand for data that is not yet opened. For example, some local groups in Finland (e.g., Helsinki and Oulu) provide the potential for organizations, companies, and individuals to demand data to be opened. They also allow users to provide feedback about the data that is already open. Thus, they meet the cyclical characteristics (Pollock 2011; Sande et al. 2013) of an open data ecosystem. In addition to data from government, institutions, and private companies, recently, different forms of social media, such as Twitter, Facebook, or Instagram, provide more and more data available online. This kind of social media data is obviously open as such, as it is based on free-formed conversations or other volunteer releases, both from communities or individuals. Due to the continuous growth of the usage of social media and the different yet increasing social media forms, the amount of this "big data" is rapidly growing. This data may not have a rational or an organized structure when compared with organizational data, but when properly treated, it can be valuable in several ways.

Open data is the main resource of the open data ecosystem. Open data and its definitions have evolved from basic definitions and principles via classifications and other kind of checklists for open data certification services (Heimstädt et al. 2014a). Figure 3 illustrates the development stairs of open data. From the first definitions, it took about 5 years before the reusability of data from the user perspective were considered. For example, the government of Great Britain proposed in 2012 a five-star scheme for assessing the degree to which the individual datasets are reusable (HM Government Cabinet Office 2012): 1 star: the data is available on the web in any format, 2 stars: the data is available in a structured format, 3 stars: the data is available in an open, non-proprietary format, 4 stars: Uniform Resource Locators (URIs) are used to identify the data using open standards and recommendations from W3C, and 5 stars: the data is linked to other people's data to provide content. A few years after that, data certification approaches emerged. For example, the Open Data Institute (ODI) provided Open Data Certificates[5] that enabled data providers to assess the extent to which open data is published according to recognized best practices. The certificate tells data users what the data is about and how to get hold of it, sharing legal (e.g., licensing, privacy), practical (e.g., discovery),

[1] https://okfn.org/
[2] http://theodi.org/
[3] http://globalopendatainitiative.org/
[4] http://inspire.jrc.ec.europa.eu/
[5] https://certificates.theodi.org

technical (e.g., structure, quality), and social (e.g., documentation) information. In the future, digital services will be able to automatically certificate open data.

Generally, an open data ecosystem consists of actors, i.e., the organizations and individuals with the roles of data suppliers/providers, data intermediaries, and data consumers (Heimstädt et al. 2014b). In addition to data and actors, the existing literature contextualizes open data ecosystems according to the following characteristics (Heimstädt et al. 2014b):

- Nested structure: The data ecosystem has a nested structure with micro, meso, and macro levels.
- Cyclical: After the data has been released, data consumers are able to view the data, edit, and update it and also contribute to it and provide their feedback (Pollock 2011; Sande et al. 2013).
- Demand-driven: The ecosystems are formed in response to the demand for data (Boley and Chang 2007).
- Sustainable: The ecosystem finds ways to emerge in the event of sudden changes (Boley and Chang 2007).

These characteristics are also essential for open data-based digital service ecosystems. The digital service ecosystems can exist on micro, meso, and macro levels, depending on the size and the amount of the value networks, and the size and scope of the provided digital services. The digital service ecosystems also implement a cyclical structure and data cycles, which enable data consumers to act as data providers, and vice versa. In a digital service ecosystem, the actors cooperate to fulfill a certain demand, and, thus, the ecosystem is demand-driven. Finally, the digital service ecosystem finds a new balance and substitutes in the event of changes. For example, new partners and data providers are sought in the case when certain data is no longer provided as open.

2.3.2 Quality of open data

A lot of work has been done to standardize quality attributes in the field of software engineering (ISO/IEC 2001; ISO/IEC 2003) and software architecture design (Gorton and Klein 2015; Immonen and Niemelä 2008; Ovaska et al. 2010; Niemelä and Immonen 2007; Kazman et al. 2000; Dobrica and Niemelä 2002). Although data quality has been the subject of several studies (Castillo et al. 2011; Agichtein et al. 2008; Gil and Artz 2007; Dai et al. 2008; Naumann and Rolker 2000; Nurse et al. 2013), the quality issues are not commonly brought into use in the case of data. The ISO 25012 data quality model (ISO 2008) defines 15 data quality attributes and classifies them into inherent quality and system-dependent quality. Some of the existing research on data quality uses the quality model as a basis, such as (Behkamal et al. 2014; Rafique et al. 2012). Data quality evaluation is challenging because data quality cannot be judged without considering the context or situation at hand (Nurse et al. 2011; Bizer 2007; Bizer and Cyganiak 2009). At this moment, there are neither agreed classifications for the applicability of quality attributes to certain contexts nor are there agreed definitions of quality attributes themselves. Quality assessment metrics are heuristics and are designed to fit a specific assessment situation (Bizer 2007; Pipino et al. 2005). Recently, the characteristics of big data, volume, variety, velocity, and veracity, have also been detected to define new challenges for data quality and data quality assessment (Ferrando-LIopis et al. 2013; Cai and Zhu 2015). Recent research on the quality of online data can be summarized

under three main factors (Nurse et al. 2011): (1) provenance factors refer to the source of information, (2) quality factors reflect how an information object fits its intended use, and (3) trustworthiness factors influence how end-users make decisions regarding the trust in the information.

The availability of the information in a machine-readable format with the commonly agreed metadata facilitates data cross-reference and interoperability and, therefore, considerably enhances the value of information for reuse (European Commission 2011). For example, data.gov.uk already includes basic metadata about all its data sets (HM Government Cabinet Office 2012). Currently, there are some de-facto standards for metadata, such as the Dublin Core Metadata Element Set (http://dublincore.org/) and the metadata of the CKAN data portal platform (http://ckan.org/). However, recent metadata standards do not assist in determining the quality of data from the data end-user's viewpoint. Different parties use different, informal ways to ensure the quality of data. For example, the ODI's Open Data Certificates rely on the data providers' assessment, enabling the users to decide how much to rely on the data.

2.3.3 Ecosystem-based digital service engineering

The digital service ecosystem takes characteristics both from business ecosystems (Zhang and Fan 2010; Li and Fan 2011; Iansiti and Levien 2004) and software ecosystems (Bosch 2009; Jansen and Cusumano 2012; Hanssen and Dybå 2012). However, in a digital service ecosystem, the service provider shares the service taxonomy and service descriptions that enable the dynamic, behavioral, and conceptual interoperability and interactions between services (Immonen et al. 2015b; Pantsar-Syväniemi et al. 2012). Just like in a business ecosystem, the members of a digital service ecosystem share the common ecosystem regulations but are able to act independently. Partner networks are created inside both ecosystems, but there are also dependencies between the digital service ecosystem members other than business dependences. Unlike in software ecosystems, in a digital service ecosystem, the members are not bound to a shared development platform or technology. However, the software can be provided as a service to the ecosystem.

Service engineering in the digital service ecosystem can be characterized according to the following features (Immonen et al. 2015b):

- Service co-innovation: open innovation enables the potential to co-create ideas for a service with other actors of other ecosystems (Stathel et al. 2008; Chan 2013; Chesbrough and Appleyard 2007).
- Service value co-creation: the value is created inside the ecosystem in value networks formed by the ecosystem members (Stathel et al. 2008; Wiesner et al. 2012) (Kett et al. 2008).
- Enabling infrastructure: the ecosystem infrastructure supports the collaboration and cooperation of ecosystem members, providing the required services and tools (Pantsar-Syväniemi et al. 2012) (Khriyenko 2012; Ruokolainen et al. 2011; Ruokolainen and Kutvonen 2009).
- Utilization of the ecosystem's assets: the existing ecosystem assets, such as the ecosystem's rules, methods, and practices for service engineering, enable co-innovation and co-creation of the services (Pantsar-Syväniemi et al. 2012; Ovaska et al. 2012; Ovaska and Kuusijärvi 2014).

Although several methods and approaches exist that take into account some of the previous features, they do not cover all of them but concentrate on their own viewpoint and not working together. Furthermore, recent approaches to ecosystem-based service engineering do not take into account the data and data quality.

3 Evolvable open data-based digital service ecosystem

This section combines and refines the earlier work of the authors on open data based business ecosystems (Immonen et al. 2014; Immonen et al. 2013), digital service ecosystems (Immonen et al. 2015b), and the quality evaluation of open data (Immonen et al. 2015a) (see Section 2.2), and it introduces the main concepts of the Evolvable Open Data based digital service Ecosystem (EODE). Interesting and certified open data is a key enabler in the EODE. Data quality certification ensures that the quality of data is verified to be good enough for the usage of the ecosystem's services. Thus, the certified data provides added value for the whole ecosystem, its members, and customers through digital services co-created based on that open data.

Figure 4 introduces the structure and the elements of the EODE; the models and services required for establishing and operating open data based service engineering (vs. Fig. 2 in Section 2). In this work, these models and services are inspected from the viewpoint of the quality of data. The term "evolvable" refers to the abilities of the digital service ecosystem to be long-lasting and to tolerate internal and external changes; the ecosystem introduces and activates survival actions based on up-to-date knowledge and support services that exploit the knowledge to adapt digital service engineering models and practices to the present situation. The EODE core illustrates a service framework that is a common infrastructure for coordinating and managing the operation of the EODE. Thus, the core contains all the mechanisms for controlling the ecosystem, including legal, practical, technical, and social aspects. In this context, the focus is on the quality certification of open data, i.e., what kinds of knowledge and support services are required from the ecosystem to ensure the quality of open data and open data services. Although the main focus is on open data and open data services, there is a brief discussion of how open data services are exploited in digital service engineering.

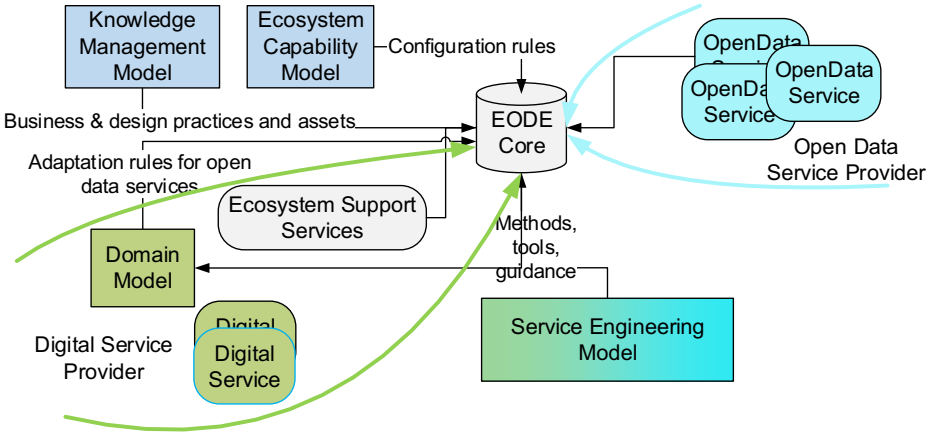

Fig. 4 Overview of an open data-based digital service ecosystem

The content of an open data based service ecosystem is specified from two viewpoints:

- *Ecosystem viewpoint* defines governance- and regulation-related actions for (a) acting in an ecosystem, (b) evaluating and monitoring the quality of open data and open data services, and (c) developing digital services on the basis of open data services. The ecosystem viewpoint has been established on the following artifacts (Fig. 4): the ecosystem capability model, the knowledge management model (KMM), ecosystem support services, and the EODE core that integrates the models and support services. The ecosystem viewpoint defines and describes how collaboration among ecosystem members is regulated, guided, and assisted. The goal is smooth collaboration among the ecosystem's members.
- *Service provider's viewpoint* defines two different viewpoints: An open data service provider's viewpoint and a digital service provider's viewpoint. The first viewpoint explains how the EODE helps open data providers to create proper open data services with the required quality. The viewpoint needs the following artifacts: ecosystem support services, the KMM, and the EODE core. The main goal is trust making among open data (service) providers and digital service providers. The digital service provider's viewpoint describes how open data services are exploited in digital service engineering. The focus is on quality evaluation of the used open data services and the digital service under development. The viewpoint exploits open data services, the ecosystem support services, the domain model, the KMM, the EODE core, and the service engineering model. The outcome is a new digital service. The main goal is to provide (personalized) digital services of high-quality.

3.1 Ecosystem viewpoint: models and services for operating in the ecosystem

This section describes the models and services common for all ecosystem members. These include the capability model, the KMM, support services, and the EODE core from Fig. 4. Figure 5 illustrates the relationships of these elements; the ecosystem support services and the knowledge management model implement the ecosystem capability model, and these are provided as services to the ecosystem through the EODE core. These elements are described in the following sub-sections from the viewpoint of the quality of open data.

3.1.1 Capability model

The capability model defines the purpose of the ecosystem, its ability to perform actions, and the rules governing how to operate in the ecosystem. The capabilities define the governance activities and regulations for directing, monitoring, and managing the ecosystem, and the activities and regulations for open data certification (including quality, availability, privacy,

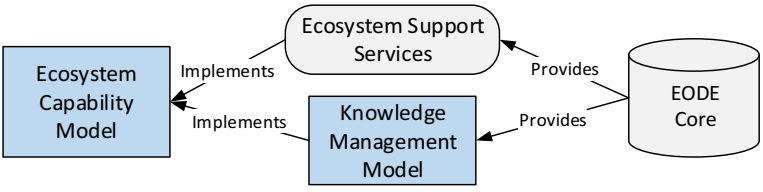

Fig. 5 The relationships of the common elements of the ecosystem

licensing) and digital service development. In this context, these capabilities will be examined from the quality evaluation of the open data and open data services perspectives.

The EODE supports community-based cooperation and collaboration among ecosystem members by providing service engineering facilities for open data service providers and digital service providers. The capabilities are implemented in the form of actions that, in EODE, are clustered according to the stakeholders' activities into three categories:

i. Quality-related activities for governance and regulation actions of the ecosystem for

- Finding reliable and trusted data sources/service providers/ecosystem members
- Contract making with ecosystem members
- The SLA specification of open data service providers and digital service providers. SLA defines the kinds of tactical rules that are used for quality evaluation. Tactical rules depend on the member's role in the ecosystem
- Supporting the bi-directional communication between digital service providers and open data service providers
- Defining an ecosystem policy that is to be followed by the ecosystem members. The ecosystem policy defines strategic evaluation regulations of the ecosystem. Examples of strategic evaluation rules are the quality criteria for open data sources
- Offering standard quality evaluation practices both for open data providers and digital service providers
- SLA contract making with open data service providers and digital service providers, i.e., defining the criteria for tactical quality evaluation
- Marketing open data services and digital services of the ecosystem

ii. Quality-related activities for open data certification that

- Find acceptable open data sources
- Extract data from different types of data sources
- Check the syntax and semantics of open data and transform them to a standard format
- Enable the quality evaluation of open data services
- Change quality policies based on changes on open data sources and/or (the quality of) open data
- Provide certification of the quality of open data services.

iii. Quality-related activities for service engineering-related actions that

- Provide reusable assets for defining data requirements with the required data quality
- Assist in matching required data quality with the provided data quality of open data services
- Provide reusable assets for quality requirements specification, quality modeling, and quality evaluation of digital services
- Test the digital services with the EODE service architecture specification
- Certify the digital services

The rest of this section concentrates on the activities of the second category since these activities guide the definition of the KMMs and support services required for achieving

certified open data to be used by the ecosystem members. The activities of the first and the third categories also influence the content of KMMs and support services and are, therefore, briefly discussed.

3.1.2 Knowledge management model (KMM)

The KMM includes common models and transformation rules for adapting specific data models to the common ones shared and accepted among ecosystem members. These models can include metadata models of (open) data, standard data models of specific application domains, and rules for how some specific data models can be adapted from a domain-specific data model to the common data model. The KMM includes the following types of quality-related knowledge:

- Ontologies that conceptualize the things related to data, quality, metrics, and services. The quality attribute ontology, e.g., reliability ontology (Zhou et al. 2011), defines the sub-characteristics of the quality attribute, metrics (García et al. 2006) for each sub-characteristic, application time, the formula used as a measuring method, value range, and target value (Immonen et al. 2015a; Niemelä et al. 2008). Context ontologies are required to identify the situation of the digital service and to carry out the situation-based service adaptation of that digital service (Pantsar-Syväniemi et al. 2011). Rules can be represented as ontologies as well.
- Design-time artifacts, i.e., architectural styles and patterns (Ovaska et al. 2010; Ovaska and Kuusijärvi 2014). It is also possible to use ontology orientation to represent the concepts of architectural descriptions and styles. In (Guessi et al. 2015), the ISO/IEC/IEEE 42010 standard of an architectural description is formalized and described as an ontology model, and further specialized to SOA architecture**.** Thus, the assumption is that integration architecture is represented as a common knowledge model shared among ecosystem members. Other common knowledge models may include service description ontologies, service component models, quality of service models, service composition models, and service community models (Aubonnet et al. 2015).
- Domain models that define domain-specific quality attributes, variations between the domain and the common model, and the adaptation rules for mapping the variable things to the context of the EODE.
- Policies used in quality evaluation and management (Bizer 2007; Rahman et al. 2011; Bertino and Lim 2010). The ecosystem policy defines a set of governance services that are common for all ecosystem members, and rules for how to configure and monitor these services. It also defines how SLAs for service providers are specified, configured, monitored, and adapted. Each SLA follows the same quality evaluation policy but is configured and adapted according to the service provider and the context of the used digital service and its user(s). Moreover, ecosystem policy also manages the following data quality policies:

  – Data filtering policy: This policy defines which open data sources are acceptable in the ecosystem. The data sources must fulfill the quality criteria of the ecosystem.
  – Data quality evaluation policy: This policy defines the quality attributes, metrics, and rules for their applicability for quality evaluation. The policy is used for data quality evaluation of the ecosystem, but it is also configurable for the specific need of each service provider.

– Decision-making policy: This policy defines how decisions are made based on the strategic or tactical operation of the ecosystem. Input for strategic decision-making is collected with the identification and analysis of changes in the ecosystem and its surroundings, e.g., related open data markets. The tactical operation of the ecosystem needs different kinds of decision-making: e.g., defining and updating actors' roles, business models, and value networks. Service engineers need online guidance realized by means of semantic wikis, a well-defined model-driven engineering environment, and continuous synchronization between the wiki and MDE-based models (Baroni et al. 2014).

3.1.3 Ecosystem support services

The purpose of the ecosystem support services is to assist in carrying out the tasks defined as activities in Section 3.1.1. The support services that evaluate the quality of open data services are common for all members. Support services provided for open data service development and digital service development are recommended, but member-specific solutions are also allowed. In that case, they need to be adapted to work in a way specified in the knowledge management model and service engineering model. In this context, the focus will be on the quality-related activities that boost open data service development and ensure the quality of open data and open data services (Section 3.1.1.ii):

A1: Defining acceptable open data sources—requires services for searching and evaluating the quality of open data sources, and monitoring the quality of data sources accepted to the ecosystem; thus ensuring that the quality remains as acceptable.
A2: Extracting data from different types of data sources—requires services for monitoring the quality of open data sources and open data, ensuring that the quality remains as acceptable.
A3: Checking syntax and semantics of open data and transforming the open data to a standard format—requires services to ensure that the data is syntactically and semantically straight, and to transform the data into the format acceptable to the ecosystem.
A4: Enabling quality evaluation of open data services—requires tailorable services that enable the evaluation of the quality of the open data service in its usage context against the required quality.
A5: Changing quality policies based on the changes of open data sources and/or (quality of) open data—requires services that enable the detection of different contexts and changes, and adapt the models and support service to the changes or in a situation-based manner.
A6: Certification of the quality of open data services—requires services to enable the validation of the open data service in the usage of the ecosystem.

The initial taxonomy of ecosystem support services (Fig. 6) defines an evolving set of services selected for the common use of ecosystem members. The ecosystem support services include eight main categories; utility services enable the management of the ecosystem policy and also the policies for data filtering, data quality, and decision-making. Matchmaking services assist in verifying the syntax and semantics of the data, thus, ensuring that the data is in the right form and is usable for the ecosystem members. Monitoring and evaluation services monitor the open data sources, the open data itself and the open data services, and

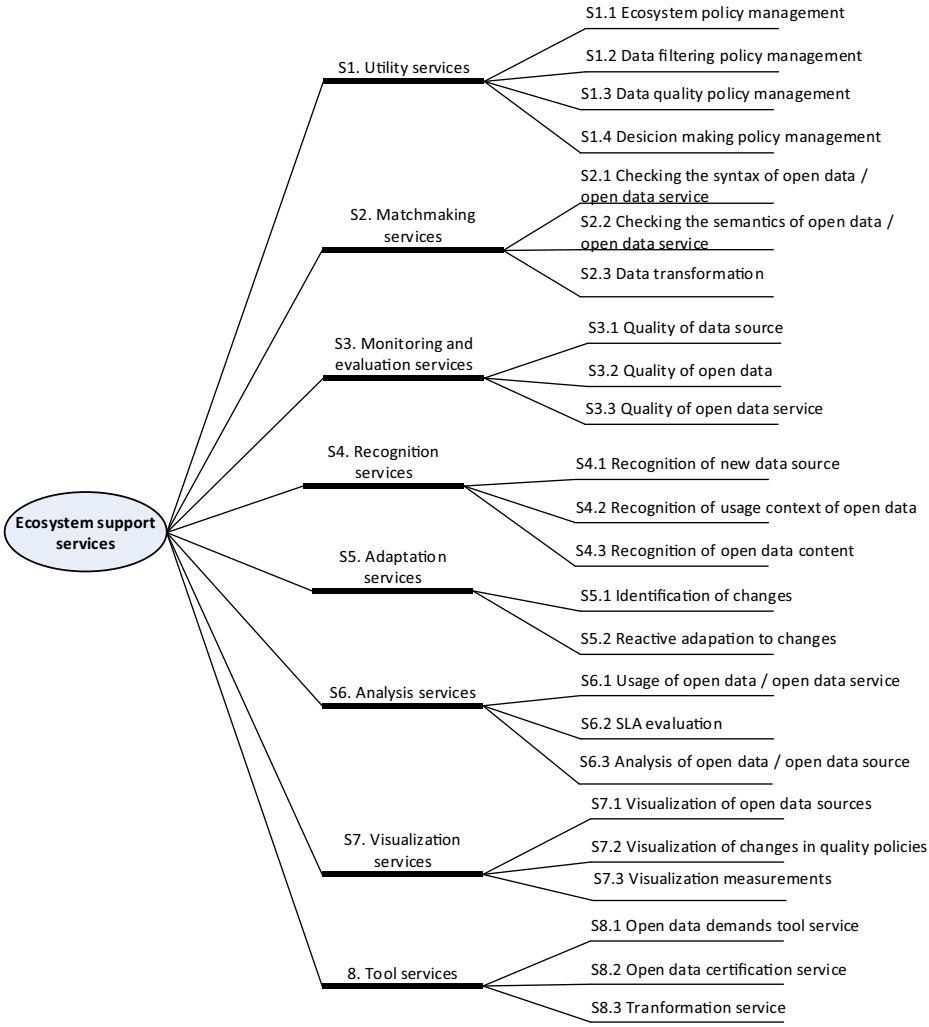

Fig. 6 The taxonomy of ecosystem support services

detect changes in their quality. Recognition services recognize the changes in the context (e.g., a new data source or changed usage context). Adaptation services adapt to the recognized changes according to policies. Analysis services perform the data quality evaluation according to the quality policies and also evaluate the SLAs between the open data service provider and the digital service provider. Visualization services provide views of the open data and open data services. Finally, tool services assist in all activities of the ecosystem members.

3.1.4 EODE core

The EODE core is an integration framework for combining models and support services for developing open data services and digital services based on them. The integration framework registers open data services, digital services, and support services and provides knowledge

management, service engineering, and domain models also as services. Thus, in addition to being an integration framework, the EODE core also acts as a means of knowledge sharing. The core also provides mechanisms to control and manage open data certification; in this context, the focus will be only on quality certification.

The core is a centralized system for maintaining a list or catalogue of the digital services of the ecosystem and additional information, such as service user feedback and rating, access management, availability information, and service logging. Service registration (see Fig. 7) is a process in which the necessary information for using and discovering the service is published in a uniform way. First, the service provider registers the services and receives a unique ID (within this registry) for the service. Second, the service provider adds the required service descriptions. At the end, the URLs of service endpoints are linked with the service resource description. Service discovery (see Fig. 7) is based on the registered service descriptions. Basic service discovery is enabled by the human readable service description and additional information associated with the service description. For more intelligent service discovery and, in particular, intelligent service matching, a semantic service data description is required. Semantically enriched descriptions support (i) multilingual searches, (ii) matching different data elements that describe the same thing, and (iii) using the relations of data elements in searches. Semantics also support interoperability between different services. These are discussed in more detail in Section 4.2.

3.2 Service providers' viewpoint: models and services for collaboration

Two main types of service providers collaborate and co-create in the evolvable open data-based digital service ecosystem: open data service providers and digital service providers. The EODE forms a two-directional communication channel between open data providers and digital service providers (Fig. 8). Trust making among service providers is supported by a common open data service model and quality assessment services provided as ecosystem support services. Due to the common knowledge management model, the service engineering model and ecosystem support services, the collaboration among diverse actors is smooth and interoperable at the levels of business, technology, and processes.

Open data service providers encapsulate their offerings (open data) with the service model and the quality policy adapted according to the current situation, thus, utilizing the knowledge management models of the ecosystem. Ecosystem support services are used in service development and in ensuring that the service is interoperable by the ecosystem members.

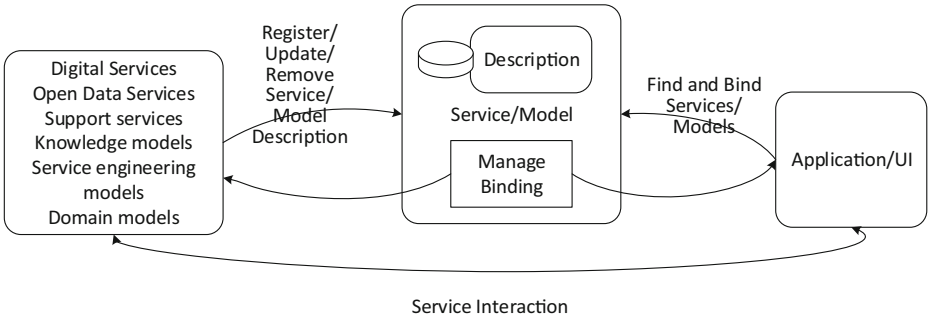

Fig. 7 Digital service framework

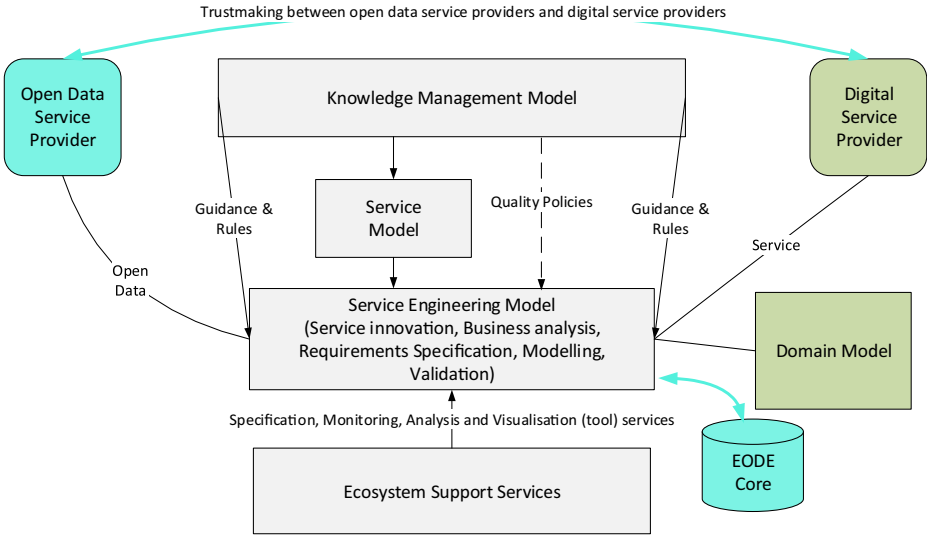

Fig. 8 Collaboration between open data service providers and digital service providers

These new open data services are provided for markets (i.e., to ecosystem members and outsiders) through the EODE core.

Digital service providers use open data services as building blocks in digital service development, and provide digital services that can be (1) domain-specific services to global markets, (2) support services to ecosystems, or (3) tool services or technology enablers for open data providers. Digital service providers utilize the support services and knowledge management models in their service development activities, and they utilize the EODE core for searching for applicable data and for registering the digital service into the EODE core registry, from where it can be searched.

Next, the service model, domain model, and service engineering model are introduced.

3.2.1 EODE service model

Open data providers encapsulate the open data and provide it as an open data service with a standard service interface, including syntactical and semantic definitions. Each open data provider can have their own data model or they can utilize the common EODE service model. The open data service interface must be implemented as a common standard, such as REST-API or/and SOAP interface.

A generic service model is defined for all kinds of digital services. The KMM can include several digital service models. A service provider can also utilize their own service model. In that case, (being an accepted member of the ecosystem), this service model can also be included as an acceptable service model for the ecosystem. The digital service model defines a common digital service interface that includes

- Interface description according to the selected architecture style, e.g., as a REST-API
- Service capabilities as a service ontology, e.g., (Kantorovitch and Niemelä 2008)
- Utility services for monitoring service availability and data quality management
- Related rules defined as policies

Ecosystem support services are internal services used by ecosystem members as part of the service engineering of digital services. The EODE core can be used for marking these digital services to customers and service users. However, other market places may also be exploited. In this context, the EODE core will be examined as a means to market open data services and digital services as well.

Open data services are categorized according to the purposes of usage and application domains. Generic open data services that can be used in any application domains include, e.g., open data from sensors and location[6] or information concerning culture and up-to-date activities.[7] Domain-specific data is categorized according to the application domain, e.g., traffic, transportation, and health.

3.2.2 Domain model

The domain model provides configuration rules for adopting open data services to match the quality requirements of the digital service under development. The digital service engineering context specifies how the open data service should be adapted. Domain-specific adaptation rules form a means to perform reactive adaptation according to the situation at hand. For example, the data format alignment service is used to adapt open data to the common data model of the ecosystem.

3.2.3 Service engineering model

The service engineering model provides the methodology and tools for developing open data services and digital services. It supports service innovation, business analysis, requirements identification, negotiation, and specification. The modeling of digital services exploits the SOA integration architecture described in the KMM and the related tool services used to describe the functional and non-functional capabilities of a new open data service or a new digital service. The service engineering model is described in more detailed in (Immonen et al. 2015b).

4 Open data quality certification process

This section describes how the elements of the ecosystem specified in the previous section are used to implement the quality certification of open data. The purpose of the certification process is to verify the trustworthiness and quality of open data, i.e., that the data comes from a reliable source, and its quality is good enough to be accepted for the usage of the ecosystem's services. The certification is a continuous process; the quality of data sources and the data itself is evaluated and monitored, and its exploitation in the ecosystem and its value is continuously evaluated. Sub-section 4.1 describes the certification process in more detail. Sub-section 4.2 provides an example of the usage of the quality policies in connection with the certification process in order to help understand how the quality certification is managed with the help of the policies in the EODE.

[6] http://www.paikkatietoikkuna.fi/web/fi
[7] http://www.hri.fi/fi/

4.1 Description of the certification process

The quality of open data is certified in five phases in the ecosystem (see Fig. 9). Some of the phases are continuous processes in the ecosystem, controlled by the ecosystem quality policies, whereas some of the phases are triggered by an event. The phases are described in the following:

P1—Acceptance: The demand for open data comes from the ecosystem. The search for new data sources is either triggered after a certain time period by the ecosystem policy or the search is triggered by an ecosystem member that demands new data. The open data source and the open data itself are evaluated to be accepted for the ecosystem with the help of the quality evaluation. The evaluation target is, first, the data source, then, the data content, and, finally, the data quality. The data extraction is a continuous process, and the quality of the (accepted) data source and the open data itself is evaluated in connection with data extraction.
P2—Adaptation: The accepted open data is modified to follow the interoperability requirements (e.g. related to format, syntax, and semantics) of the ecosystem. Thus, the open data is transformed or adapted to an open data service that can be used as a building block in digital services.
P3—Validation: The digital service provider validates the open data against its intended use, i.e. whether the data is fit for use within the certain context and situation of the digital service provider. The service provider configures the quality evaluation policy according to its own organizational data policy.
P4—Exploitation: The open data sources and the usage of the open data are monitored, and feedback from users is collected. The users of the data are the digital service providers that use the data in their services, and the service consumers that utilize the data via digital services. The ecosystem enables reactions to changes and allows decision-making, based

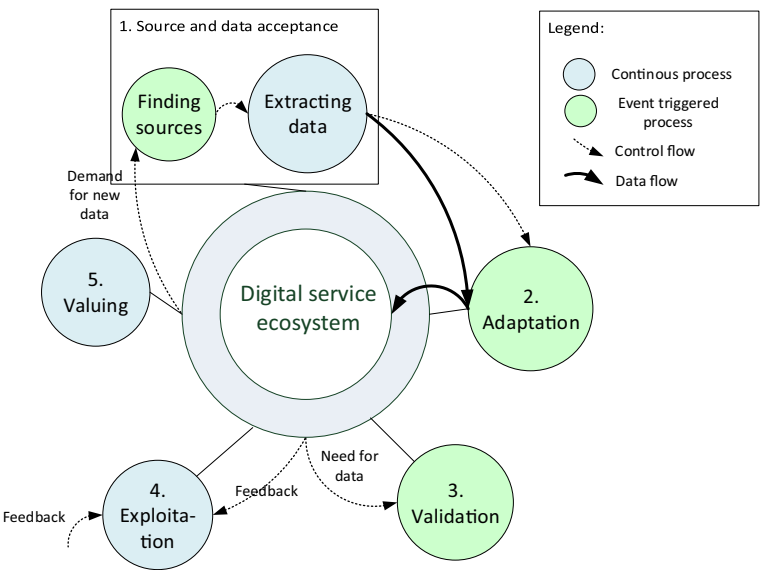

Fig. 9 The phases of the open data quality certification process

on the collected open data, by visualizing the alternatives and enabling the configuration of parameters.

P5—Valuing: The quantified value of the open data service is continuously evaluated. This includes the value comparison of open data services and the decision to keep the service or substitute it with another service.

Table 1 describes the quality attributes with the metrics and measurement approaches that were identified to be applicable in the ecosystem context. The measurement approach is defined as a sequence of operations aimed at determining the value of a measurement result, being a measurement method, a measurement function, or an analysis model (García et al. 2006). The measurement method is a logical sequence of operations that is used to quantify an attribute with respect to a specified scale (defining a base measure). The measurement function is an algorithm or calculation performed to combine two or more base or derived measures (defines a derived measure). The analysis model is an algorithm or calculation that combines one or more measures with associated decision criteria (defining an indicator). The quality attributes are defined by the knowledge management models, i.e., policies, and are evaluated with the related support services.

Table 2 maps the activities (A1–A6 described in Section 3.1.3) to the certification phases (P1–P5), and summarizes the support services and knowledge assets that are required to implement each activity. Table 2 also maps the derived quality attributes for evaluating the quality of open data/open data services to each activity.

Data filtering, data quality evaluation, and decision-making policies have different purposes in each evaluation phase in the ecosystem. Open data can originate from different kinds of source types, and each type can have different kinds of properties relating to, for example, the data content, structure, and size. Therefore, the first thing that the data filtering policy must define is acceptable data source types. Each data set is then classified into these types. The open data source types can be, for example, web pages (free-formed), Facebook, Twitter, Instagram, customer feedback, analyses and reports, or other semi-structured documents. The filtering policy and data evaluation policy must define the quality properties and rules specific for each data source type for data quality evaluation. These include the following:

- The attributes for data source/open data/open data service evaluation
- The metrics of which the attribute consists and which are used in the assessment
- The value range for each metric
- The formula for achieving the metric value from measured value
- The acceptable value for each metric
- The rules that define which attributes/metrics are taken into account and which weights are assigned to the metrics

The filtering policy uses the quality metrics and rules in evaluating whether or not to accept the data set to the ecosystem. Thus, the data filtering policy defines the quality criteria for open data sources, i.e., the strategic evaluation criteria. The data evaluation policy is used by the ecosystem in data extraction, in data monitoring, and in decision-making. Also, service providers have their own evaluation policies when searching data for a certain purpose (in phase 3). The quality evaluation policy is utilized as criteria for SLA specification and tactical quality evaluation of the data itself. The decision-making policy defines the criteria for actions based on quality evaluation, such as how to adapt to changes or what actions to take based on evaluation results.

Table 1 Metrics and measuring methods used in quality attribute evaluation

Quality attribute	Description	Metric	Measurement approach
Believability	The extent to which data is regarded as true and credible; credibility of the data source, comparison to a commonly accepted standard, and previous experience (Pipino et al. 2002)	The lowest number of three properties; the believability of the data source, believability against a common standard, and believability based on experience. E.g., the source has a verified account.	Analysis model
Completeness	The degree to which data is not missing; schema level: all of the required classes and properties are represented, data level: no missing values of properties with respect to the schema (Behkamal et al. 2014)	E.g., completeness of data within a data set (A/B); A = number of data required for the particular context in the data set, B = number of data in the specified particular context of intended use	Measuring function
Consistency	Consistency of the values of data; implies that two or more values do not conflict with each other (Mecella et al. 2002) (Wand and Wang 1996)	Data consistency checks	Measuring function
Corroboration	The same data comes from different sources (Dai et al. 2008)	The amount of data sources	Measuring function
Coverage of data	The extent to which the volume of data is appropriate for the task at hand (Pipino et al. 2002)	The minimum of two ratios: The ratio of the number of data units provided to the number of data units needed, and the ratio of the number of data units needed to the number of data units provided	Measuring function
Popularity	The data has a number of followers, and/or the data is liked and repeated by others	Number of data users; Number of re-tweets	Measuring function
Relevancy	The extent to which the data is applicable and helpful for the task at hand (Pipino et al. 2002)	The number of occurrences of relevant key words	Measuring function
Semantic accuracy	Data entities must reference a real world correspondent and must have faultless attribute values (Behkamal et al. 2014).	Detection of outliers; Detection of inaccurate values	Measuring function, Analysis model
Syntactic accuracy	Structural validity of a dataset, such as compliance with the RDF/XML standard (Behkamal et al. 2014)	E.g., A/B; A = number of records with the specified field syntactically accurate, B = number of records	Measuring function
Timeliness	The freshness of the data	A timestamp: data set creation date	Measuring function
Uniqueness	The non-redundancy characteristic of the entities, classes, properties, and values of properties in a dataset		Analysis model

Table 1 (continued)

Quality attribute	Description	Metric	Measurement approach
	(Behkamal et al. 2014)	Number of redundant data classes; ratio of similar properties; ratio of redundant instances; ratio of functional properties with different values	
Validity	The likelihood that the data in an appropriate format and the values are still valid (Ramaswamy et al. 2013)	Data syntax and semantic checks; Data set creation date	Analysis model
Verifiability	The degree and ease with which the data/information can be checked for correctness. E.g., checking the data source names from the actual source of the data, or whether the data points to a trusted third party source where the data can be checked for correctness (Naumann 2002)	Resource identifier, Relation, Cross references	Analysis model

Table 2 Mapping between the activities, support services, knowledge assets, and quality attributes

Phase	Activity	Support services	Knowledge assets	Quality attribute	Evaluation target
Acceptance (P1)	Finding out relevant open data sources (A1)	Recognition and acceptance of new data sources (S4.1) Content recognition of open data (S4.3) Quality evaluation of open data source (S3.1)	Filtering policy	Believability Popularity Verifiability Timeliness	Data source
	Extracting open data from different types of data sources (A2)	Quality monitoring of open data sources (S3.1) Content monitoring of open data(S4.3) Quality evaluation and monitoring of open data (S3.2)	Quality evaluation policy	Believability Popularity Verifiability Timeliness Uniqueness Coverage of data	Data source, open data
Adaptation (P2)	Checking syntax and semantics of open data and transforming these to a standard format (A3)	Checking syntax of open data (S2.1) Adaptation of open data to the standard format (S2.3) Checking semantics of open data (S2.2) Semantic alignment of open data (S2.3)	Service model	Syntactic accuracy Semantic accuracy	Open data
Validation (P3)	Enabling quality evaluation of open data services (A4)	Recognition of usage context of open data (S4.2) Quality evaluation of open data services (S3.3)	Quality evaluation policy	Validity Completeness Relevancy	Open data
Exploitation (P4)	Changing quality policies based on the changes of open data sources and/or (quality of) open data (A5)	Monitoring the quality of open data sources and open data (S3.1, S3.2) Monitoring the use of open data (S6.1) Analyses of open data sources and (quality of) open data (S6.3) Visualizing quality measurements (S7.3) Configuring quality policy (S1.1–S1.4) Visualizing changes in quality policy (S5.2, S7.2)	Quality evaluation policy Decision-making policy	Popularity Corroboration Coverage of data Relevancy Validity	Open data service
Valuing (P5)	Certifying open data services (A6)	Checking syntax of open data services (S2.1) Checking semantics of open data services (S2.2) Quality evaluation of open data services (S3.3)	Service model Quality evaluation policy	Syntactic accuracy Semantic accuracy Uniqueness Completeness Consistency	Open data service

The policies are expressed using event-condition-action (ECA) rules. ECA rules take the form "when Event occurs and Condition holds, then execute Action," in other words, the ECA rules are composed of event definitions, triggering conditions, and the actions to be taken.

4.2 An example of the usage of policies in data quality certification

Table 3 provides an example of a detailed description of how policies and support services are related to the two activities of phase 1 of the certification process. Data filtering policy enables the selection of reliable data and data sources for the ecosystem. The description of the policies of the example concentrates on the data source type "Twitter." Table 3 describes two activities. Activity 1 is described in more detail below.

4.2.1 Introduction of policies

According to Table 3, the data filtering policy is used in phase 1, activity 1 ('Finding out relevant open data sources'). The content of the policy is described in more detail below:

A) *Acceptable data source types and the acceptable content for each data source type*: The filtering policy defines the list of acceptable data source types and the content for each data source type, e.g.,

$$Data\ source\ type = Twitter, content\ type = tweet$$

$$Data\ source\ type = Youtube, content\ type = videos$$

$$Data\ source\ type = Facebook, content\ type = text, pictures, videos$$

Table 3 The policies and support services that implement the activities

Policy	Policy content	Support services
Phase 1 activity 1: finding out relevant open data sources		
Data filtering policy	A) Acceptable data source types and the acceptable content for each data source type	Content recognition (S4.3)
	B) Quality attributes and evaluation metrics for each data source type	Quality evaluation of data source (S3.1)
	C) Value range and rules for acceptance for each data source	Data source recognition service (S4.1)
Phase 1 activity 2: extracting data from different types of data sources		
Data filtering policy	Description of acceptable data sources	Data source recognition service (S4.1)
Quality evaluation policy	Definition of quality attributes and metrics for each data source type	Quality evaluation of open data (S3.2)
	Definition of rules for data set acceptance	

B) *Quality attributes and evaluation metrics for each data source type*: Table 4 describes an example of the attributes, metrics, and their value range included in data filtering policy for the data source type "Twitter".

C) *Value range and rules for acceptance for each data source*: The following describes the filtering policy rules for the data source type "Twitter":

Event: Finding out relevant open data sources
Condition:

IF the data source type = *Twitter* AND *the content type* = *tweet* AND
IF (*Data publication date* > 1.1.2015) AND (*Popularity* > 0.6 *OR Believability* > 0.9) AND *Verifiability* = 0.7

Action: The data source is assigned as an acceptable data source for the ecosystem.

4.2.2 Usage of the policies

Figure 10 illustrates how support services exploit policies in phase 1, activity 1 and in phase 1, activity 2. The usage of the policies is described in more detail below:

Table 4 An example of the elements of the data filtering policy

Attribute	Metric	Formula	Value range (% of the estimated maximum value of the topic; normalized between 0…1)	Acceptable value
Timeliness	Data publication date	Retrieving the date	1.1.2015–today	Newer than 1.1.2015
Popularity	Number of followers in a day	Followers count	<0.2 = "not relevant"; <0.4 = "not acceptable"; ≥0.6 = "acceptable"; ≥0.8 = "relevant"; ≥0.9 = "highly relevant"	≥0.6
Believability (of source)	Registration age of the author in days	Retrieving registration age	<0.3 = "not relevant"; ≥0.5 = "acceptable; ≥0.8 = "relevant"; ≥0.9 = "highly relevant"	≥0.6
	Author frequency: the number of tweets/ comments/questions at posting time	Counting all tweets/ comments/questions	<0.4 = "not relevant"; ≥0.5 = "acceptable; ≥0.8 = "relevant"; ≥0.9 = "not relevant"	≥0.4 and <0.9
	The source's account is verified	Is verified	1 = yes; 0 = no	1
Verifiability (of content)	Resource identifier	Resource identifier search	1 = yes; 0 = no	1
	Number of cross references	Cross references search	<0.4 = "not acceptable"; 0.4 = "1"; 1.0 = "estimated max"	≥0.4 ≤ 1.0

Phase 1, activity 1: finding out relevant open data sources The ecosystem evaluates whether or not the new data source can be accepted to the ecosystem. The trigger for this activity can come from an ecosystem member, or the activity can be triggered after a certain time period according to ecosystem policy.

1.1. The user (or the search monitor) that follows the ecosystem capability model/ecosystem policy wants to add a new data source to the ecosystem. The user defines the data source to the EODE system through a user interface.
1.2. The recognition service checks the content of the data.
1.3. After the content has been verified, the recognition service notifies the monitoring and evaluation service to evaluate the quality of the data source.
1.4. The monitoring and evaluation service identifies the data source type, and it evaluates the quality of the data source utilizing the data filtering policy. The data filtering policy defines the quality attributes and metrics for the data source type at hand.
1.5. The monitoring and evaluation service returns the evaluation results to the recognition service.
1.6. The recognition service checks the value range for the quality attributes and compares them with the evaluation results. The value ranges are defined in the data filtering policy.
1.7. The recognition service returns the decision whether to accept or reject the data source. In this case, the data source is accepted to the ecosystem.

Phase 1, activity 2: extracting data from different types of data sources The data is brought to the ecosystem from an acceptable data source. The data is evaluated at the time of extraction.

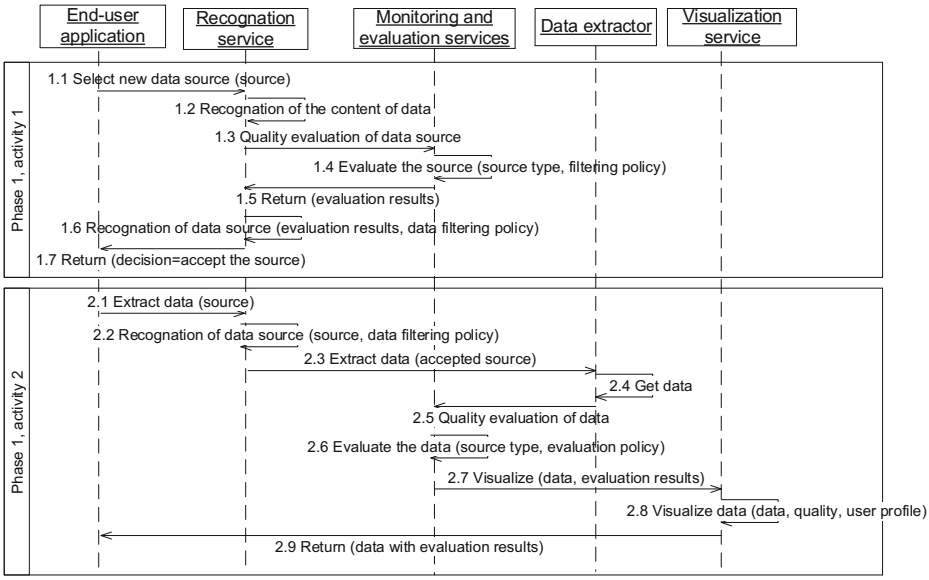

Fig. 10 The usage of quality policies at run-time

2.1. The user/monitor wants to extract data from a data source and to see the quality of the data.
2.2. The recognition service checks with the data filtering policy whether the source is an acceptable source for the ecosystem.
2.3. If the data source is accepted by the ecosystem, the recognition and adaptation service permits data extraction service to extract the data.
2.4. The data is extracted to the ecosystem.
2.5. After the data extraction, metadata is created (beyond the scope of this paper, see Immonen et al. 2015a) and the evaluation and monitoring service is requested to evaluate the quality of the data set.
2.6. The monitoring and evaluation service evaluates the quality of the data set according to the quality attributes and metrics defined in the quality evaluation policy for this kind of data source type. (Some of the quality attributes that have already been evaluated in the case of activity 1 are now reevaluated at the time of extraction.)
2.7. The visualization service is requested to illustrate the data with its quality for the user.
2.8. The visualization service illustrates the extracted data with its quality attribute values for the user according to the user profile.

5 Analyses and discussion

This section describes the maturity analyses of the main elements of the EODE concept and discussion of ongoing and future work.

5.1 Concept development and validation

In this paper, all our earlier works related to open data-based ecosystems, service ecosystems, and data quality evaluation have been combined, adapted, and extended; the concept of an evolvable open data-based digital service ecosystem is introduced. The concept specifies the capability model with the required support services and knowledge models to implement the actions related to the quality certification of open data. The development and validation of the EODE concept has been carried out incrementally in several international and national research projects. The research described in this paper was conducted in co-development in the ODEP, N4S, and DHR projects in 2015–2016. The validation of the work remains in progress in the N4S and DHR projects until 2017. The development and validation of the parts of the EODE concept are described in the following sub-sections, including the status and the maturity of the evaluation.

5.1.1 Main elements of the service ecosystem and ecosystem-based service engineering model

The digital service ecosystems were researched in Innovative Cloud Architecture for the Real Entertainment (ITEA2-ICARE)[8] project during in 2011–2014. The term digital service

[8] https://itea3.org/project/icare.html

ecosystem was relatively new at that time and not properly defined, and, therefore, a comparative definition of the properties of the business ecosystem, digital service ecosystem, and software ecosystem were first presented. Based on this state-of-the-art review, it was detected that methods for how to take the digital service ecosystem elements into account in service engineering were missing. The main requirements for ecosystem-based digital service engineering were identified, and the main elements of a digital service ecosystem and an ecosystem-based service engineering model were specified (Immonen et al. 2015b). The service engineering model included a requirements engineering (RE) method for digital service ecosystems, and it included two document templates for requirements elicitation and identification and for communication, knowledge sharing, negotiation, and decision-making. The RE method was validated in use with the ecosystem concept in two different ecosystems. The first case took place in the ITEA2-ICARE project, when the ecosystem concept and the RE method were applied to specifying the digital services and related support services for an interactive multi-screen TV services ecosystem. The goal for applying the RE method was to collect and analyze requirements from the ecosystem members for a shared service-oriented platform, which would enable the provisioning, integration, and use of services among the members of the ecosystem. The second case took place in the Connecting Digital Cities (EU-EIT-CDC)[9] project, in connection with the open service platform offering open real-time data from several data providers. The goal of the RE method application was to extract high-level user and business requirements for the open real-time data platform. Altogether, the method was used by 32 European partners that collected 298 requirements, including functional, non-functional and business requirements, and constraints (Immonen et al. 2015b). A feedback collection among the partners that were involved in the requirement engineering in the ICARE and CDC projects was performed to obtain user experiences and opinions about the ecosystem concept and the method and to find out any advantages, shortcomings, and development targets (Immonen et al. 2015b). The RE method was seen as valuable and useful in the beginning of the service engineering process when the long-term development of new service architecture was started for digital ecosystem-based services. The service RE method was especially useful for describing, documenting, and communicating the capabilities of the digital services and the service architecture they required. The method was also seen as useful in the analysis phase, where the different stakeholders work together. However, the definition of quality requirements was identified as a development target; special skills and knowledge on quality attributes are required and should be present in the innovation and requirements analysis and in the negotiation and specification phases.

5.1.2 Concept of open data-based business ecosystem

In our initial research on open data (Immonen et al. 2014; Immonen et al. 2013), the first draft of an open data ecosystem was defined from the business viewpoint. The work was performed in 2012–2014 on the national strategic research project, ODEP (Open Data End-user Programming), funded by the Finnish Funding Agency for Technology and Innovation (TEKES) and VTT Technical Research Centre of Finland. The purpose of the ODEP project with the research theme "Open data and analytics" was to create new technology and business potential utilizing open data. The subject was, at the time, relatively new and the utilization of open data in business by companies was at the outset. The requirements of such an ecosystem were

[9] EIT ICT Labs project No. 14465

collected with the help of interviews of industry representatives and the motives and challenges of acting in the open data ecosystem were identified. Altogether, 11 industry representatives participated in the interviews, including ecosystem actors such as data providers, application developers, infrastructure providers, and application users. Companies were selected from different application domains to be interviewed, and they differed in company size and service types. The interviewees, for example, product developer managers, customer and development managers, and finance and administration managers, were selected based on their knowledge of the business viewpoint of their company. The interviews provided valuable insight and requirements for the concept of an open data-based ecosystem and enabled a response to the actual needs of the data-based industry. Furthermore, the interviews enabled identifying the challenges and opportunities of open data, and applications and services of open data, and enabled evaluating the feasibility of the open data ecosystem (Immonen et al. 2014).

5.1.3 Solution for quality evaluation of open data

The evaluation of the open data quality was the main concern in the work in (Immonen et al. 2015a), in which the elements and phases of quality evaluation of open data in big data architecture were defined. The research was conducted under DIMECC's Need for Speed (N4S) program[10] funded by TEKES, in 2013–2016 and will continue until 2017. A solution for quality evaluation of open data was developed, which based on data quality policies, defines the evaluation metrics, extracts and evaluates data from Twitter, and visualizes the data for users weighting the relevant quality attributes of the user. The solution was validated with the help of an industrial case example; the solution provided a major data consulting company insight into customer needs, facilitating the R&D of the company. The solution evaluated the quality and trustworthiness of data and, thus, provided verified data for the company's business decision-making. The data in the case study was social media data (mainly from Twitter), but the approach can be extended to other data source types as well. The validation showed that although the evaluation succeeded within the case example, much more work remained to be done to extend the solution to be applicable to other data source types as well.

The current implementation of the quality evaluation of open data supports quality evaluation inside a single company; the quality policies must be implemented to be applicable to the ecosystem context. In EODE, the evaluation is done with the help of quality policies on two levels; ecosystem and service providers. The purpose of the quality evaluation performed by the ecosystem is to ensure that the quality of data is good enough to be accepted to the ecosystem. The purpose of the quality evaluation performed by the service provider is to ensure that the data fits the intended use of the provider. The service provider must first identify the intended use of data. This should be defined in the company's strategy. The ecosystem assists the service provider in specifying their own quality policies. After that, the service provider evaluates the data quality with the help of quality attributes and metrics applicable to their own context. These should be defined in the company's own quality policies. The service provider does not necessarily have to know anything about data quality evaluation methods or techniques. The provider can specify their data requirements with the help of the evaluation policy "template" provided by the ecosystem. The "template" of policy assists in selecting the evaluation attributes applicable, and, depending on the type of data source, the applicable metrics and techniques can then be selected automatically. The

[10] http://www.n4s.fi/en/

evaluation can also be automated, i.e. the service matchmaking algorithm can perform the quality evaluation with the help of the quality policy defined by the service provider.

5.1.4 Semantic data model

Previous attempts at the application of semantic data structures to the presentation of data, and thus bringing necessary understanding to the data, have not become widely adopted because of the additional work required. The presentation of data semantics would require modification not only to the data structure itself but also to all the applications that produce and use the particular data. To tackle this problem, the authors have made several contributions to data semantics. In (Pantsar-Syväniemi et al. 2012), a generic adaptation framework for developing situation-based applications for smart environments is described that embodies a novel architecture and general ontologies that solve the semantic, dynamic, behavioral, and conceptual interoperability problems of most physical environments. The semantic models (in the form of ontologies) ensure interoperability beyond communication and the interoperability of the information exchanged, the interoperability of context and its changes and the interoperability of application behavior. The applications developed based on the framework can use and apply semantic information in different kinds of smart places (for example, in homes, offices, and cities). A presentation of how the applications are developed based on the ontologies is provided in (Ovaska and Kuusijärvi 2014), in which the run-time quality adaptation is also described. The developed approach was applied to the development of a semantic facility data management system (Niskanen et al. 2014) that was incrementally developed and validated with four industrial pilots that were carried out in 2011–2014. The semantic models included domain-specific parts, which complicated their usage in different domains, but the security and context ontologies were generic and applicable to any domain.

Commonly, open data is published as a REST (REpresentational State Transfer)[11] resource without any description of the data structure. Thus, the content and structure of the data remain unknown, complicating the utilization of the data. Even in the cases when a data structure is presented in a commonly accepted way, e.g., XSD,[12] the XML Schema definition, the nature of the data and the purpose of the attributes cannot be detected. As a response to this problem, the service data description with the semantic data model was developed in the Digital Health Revolution (DHR)[13] project funded by TEKES in 2015. The service data description of the model enables linking the service data description to commonly available schematics, e.g., schema.org or domain-specific ones, or optionally to a service-specific dictionary. The solution allows the presentation of heterogeneous data from different domains using a common description format that still allows the presentation of the domain-specific semantic information. Linkage to commonly available schemas minimizes the need for service-specific ontology development. The structure of the data remains unaffected, which allows the data to be presented using multiple schemas, e.g., linking to different vocabularies depending on the attribute. This kind of data description model is an efficient tool for relevant data discovery; the data search can be targeted directly to the service data descriptions using the terminology of the schema. For example, the search can be targeted to find all health data that provides continuous heart rate information. Different data structures from different data services

[11] http://www.service-architecture.com/articles/web-services/representational_state_transfer_rest.html
[12] http://searchsoa.techtarget.com/definition/XSD
[13] http://www.digitalhealthrevolution.fi/

that all provide the same information, e.g., the heart rate, are linked to the same concept regardless of the parameter names in the original data structure and the presentation method of the data. To aid the creation of the open data service descriptions, a service framework implementation[14] provides graphical tools for service registration. The digital service registry implementation[15] provides the service description database, related REST-interfaces, and data models as open source implementation.

In summary, interoperability on the data level is achieved by the service data description, dataset descriptions, and their relations to the Concept Schema and the concepts defined in it. Service level interoperability is realized by using common service interface descriptions based on the REST architecture style. The further development of the semantic data model is still ongoing in the DHR project, and will continue until 7/2017. At this moment, tools for creating the data descriptions and linking to applicable dictionaries, and service matchmaking, based on the data descriptions, are in progress. Online testing environment deployment is also ongoing with project partners, where data and service providers can publish data sources and services that produce additional value from data to service consumers. The tool for creating a semantic data description for a service uses the normal data model as input, asks the service developer for the parameters, e.g., heart rate or pulse, and possible links to domain-specific schemas and, thus, transforms the non-semantic data model to the semantic data model.

5.1.5 EODE core

The core is based on the authors' work on a Digital Service Registry. The Service Registry is a centralized system for maintaining a list or catalogue of any digital services reachable through a URL and additional description information, such as technical and human readable descriptions, location (geographic and endpoint URL), service user feedback and rating, access management, availability information, and service logging.

The EODE core, and the Digital Services Hub as its initial implementation, was developed within the scope of the ITEA2-ICARE project. The implementation was done completely using a model-based software development method. An instance of the core was published as an open service platform[16]—Digital Services Hub—that is free to use for research and innovation purposes. The Digital Services Hub was used and demonstrated in the ITEA2-ICARE project, where project partners registered their services and used the Digital Services Hub for authorizing and visualizing service connections. The Digital Services Hub provides a user interface with which the service providers can register their services. The Digital Services Hub fulfilled its purposes well in the multimedia domain of eight international service providers. However, the context was closed, and, therefore, more validation is required. Further development of the Digital Services Hub is in progress in the N4S and DHR projects. Practical experiences of the applicability of the knowledge management and capability models in digital service ecosystems are especially required.

The Digital Services Hub can contain any kind of digital entities that have a digital API, e.g., open data, support services, and digital services. The service description can include information about the service characteristics, such as the throughput and latency, with which the service consumer can evaluate whether the service is working correctly and is good enough

[14] www.digitalserviceshub.com
[15] https://github.com/digitalhealthrevolution/serviceregistry
[16] https://www.digitalserviceshub.com/registry/

for the consumer's purposes. The Digital Services Hub also handles service availability in the following ways: (1) the reported availability; the registered service notifies the services Digital Services Hub regularly when they are active. (2) Requested availability; the Digital Services Hub continuously queries its services for availability and represents the status to the service users. The service provider must implement the management API that responds to the queries.

The trust between services and data privacy is implemented in the Digital Service Hub in the context of personal data; MyData. MyData is a service in which own personal data is provided in an exploitable format. Trust and privacy management is implemented on two levels (see Fig. 11):

> *Level 1*: *The trust making*. The Digital Services Hub manages the trust between two digital services by granting a binding key between two services, e.g. service A can use service B (Fig. 11). The key ensures that the service is a recognized and trusted service, and both services are controlled though the service registry.
> *Level 2*: *Consent management*. The data owner consents to whether or not service B can use their personal data, and also consents to whether or not their data can be transformed over the binding between services A and B. By giving consent to the binding, the data owner also must accept the terms of data usage of service A.

The implementation of consent management is a part of the ongoing work with the test environment deployment of the DHR project. The data owners can give consent about the usage of their personal data using the MyData dashboard UI (see Fig. 11).

5.2 Limitations, open issues, future research items

The validation of the last two phases of the data certification process; "Exploitation" and "Valuing," requires a real environment. These two phases are the focus of future work: an ecosystem of trusted, interoperable, and legally compliant cloud services will be set up, and the focus will be on developing the required mechanisms to register, discover, compose, use, and assess the services. The Digital Services Hub framework may be used as the core of the ecosystem to register and monitor services, but it must be extended to monitor more quality attributes. The quality evaluation approach (Phase 1) must be adapted to work in the connection with the Digital Service Hub in the way that they can utilize each other.

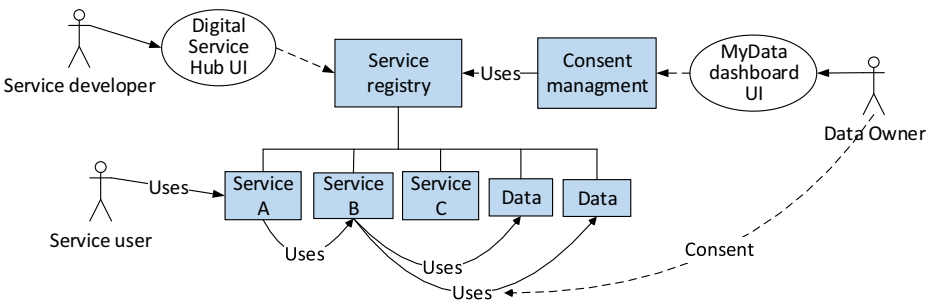

Fig. 11 The trust making and privacy management in the EODE core

Data quality was detected as the most important issue for data utilization (Immonen et al. 2014), and, therefore, in this work, data certification focused on data quality; the evaluation targets were, first, the data source, then, the data content, and, finally, the data quality itself. However, when the data quality has been ensured to be good enough, there are also many other issues that affect data selection and utilization, such as the data licenses that must be evaluated and selected. Originally, the licenses grant the "baseline rights" to distribute copyrighted work, and most licenses still contain some elements that restrict the utilization of data, such as Attribution, Non-Commercial, No-Derivatives, and Share Alike.[17] The different restricting elements can be mixed and matched, and, therefore, a huge amount of customized licenses exist for data. The selection of a license must, therefore, be done carefully and must be applicable to the situation at hand.

On the ecosystem level, there are two ways to deal with the data licenses. When the data is verified for the ecosystem based on data quality, the ecosystem can

1. Accept all data without considering the data licenses
2. Restrict data acceptance for the ecosystem based on the licenses of the data, e.g., data with certain licenses are accepted w.hile others are rejected.

In both cases, the final license selection is the responsibility of the data user, i.e., the service provider. The data provider may provide several licenses for the data. Furthermore, the service provider may utilize data from several sources in the same service. The ecosystem must bring all the license options available to service providers and assist in license selection. The service provider must be able to compare the licenses and to select the most applicable one. In the event that data from several data sources is combined to the same service, the service provider should be explicitly informed about the licenses attached to each data and the cumulative effect of the all the licenses merged together. Some assistance for the license selection has been provided for data providers. For example, Creative Commons currently provides two methods for integrating license selection into applications: the Partner Interface and the web service API. In (Daga et al. 2015), an ontology-based tool is presented for a data provider to select the license for their data. However, the selected licenses must obey the wish and intention of the data owner who, in the end, owns the rights to the data (Boris et al. 2016). In the same way, a "tool" is required for data users to compare and select licenses and to integrate several licenses of different data. This tool, "a license selection tool," could be provided as a support service of the ecosystem for service providers.

The aim is to extend the EODE with the other aspects of data certification; legal aspects, i.e., license selection and data privacy, and practical aspects (the availability of open data, open data services, and support services). These are our upcoming research topics to be examined in the upcoming international project scheduled to begin in the autumn, 2017. The data privacy aspects are domain-specific and often regulated differently in each country. Final license selection is the responsibility of the service provider; the ecosystem only assists in license comparison, selection, and integration. The concept of the EODE will be refined to include services and knowledge management models to assist the service provider in data certification based on legal aspects. Furthermore, the ecosystem's filtering policy will be used to ensure that the licenses applicable for the data are applicable also for the ecosystem. The data filtering policy may contain restrictions on license conditions that may prevent data selection for an

[17] https://creativecommons.org/licenses/

ecosystem even if the quality of the data has been ensured to be good enough. The domain knowledge model must include knowledge of how to take data privacy issues into account separately in each domain, e.g., healthcare, traffic, or financial. Thus, when applying the EODE in different domains, privacy issues must be solved case by case. The work on transforming the non-semantic data model to semantic data model is still ongoing in the DHR project, and the aim is to continue with the research topic in other research projects (not yet defined). In the next step, the concept schema will be extended and linked with the license ontologies in the same way that it is already linked with the data ontologies.

Furthermore, the transformation of the solution of the data quality evaluation to the ecosystem context is work the authors plan for the future, and it will be the focus of the upcoming international project planned to start in 2017. The quality policies will be refined to be applicable in the ecosystem context, both for evaluating and accepting data for an ecosystem, and for enabling users to configure policies specific to their own purposes.

The generic EODE concept described in this paper can be applied to certain domains when content of the ecosystem elements is to be specified on a more detailed level as the domain of the ecosystem becomes known. The domain model specifies the domain/application-specific knowledge used together with the generic knowledge management models to adapt the service engineering and digital services to the case at hand, e.g. to the healthcare, energy, and traffic domains. The generic EODE concept with the KMM and service engineering models, however, assists in defining these domain-specific models. The application of the EODE to different application domains and business fields is naturally the next step in the validation of the concept.

6 Conclusions

Poor and unknown quality has widely been recognized as one of the major obstacles for open data utilization. The main motivation of this paper is how to guarantee the quality of open data in the service ecosystem context. This paper combined the authors' earlier research on open data ecosystems, data quality evaluation, and service engineering in a digital service ecosystem, and it introduced the concept of an Evolvable Open Data-based digital service Ecosystem (EODE), which defines the kind of knowledge and services that are required for validating open data in digital service ecosystems. Open data is brought into EODE as an open data service that encapsulates the open data for the usage of the digital service developers. This paper introduced the main concepts of the EODE; the capability model that provides activities for quality evaluation of the open data, and the knowledge management models and the ecosystem support services that support these activities, thus enabling the quality evaluation of the data source, the open data itself, and the open data service. The EODE concept is general and applicable to any domain through domain models that describe the domain-specific concepts and rules.

This paper also introduced an open data certification process, which implements data quality certification with the help of EODE's knowledge management models and support services. The open data quality certification consists of five phases, which enable bringing validated open data to the ecosystem from trustworthy sources, transforming it to the acceptable form of the ecosystem, validating it against its intended usage of each service provider, monitoring the data sources and the usage of the data, and continuously evaluating the quantified value of the open data service. Some of the phases are continuous processes controlled by ecosystem quality policies, whereas some of the phases are triggered by an event in the ecosystem. The certification process is generally described, and must be adapted

and specialized using the domain model that defines domain- and application-specific extensions, replacements and adaptation rules, and regulations. Although several validation experiments of the EODE elements have been carried out during the last three years, the whole EODE concept still requires more experimental tests in different application and business fields in order to guarantee that generic and domain-specific knowledge can be maintained separate but smoothly exploited together using the generic capability and knowledge management models during the run of digital services.

References

Agichtein, E., Castillo, C., Donato, D., Gionis, A. & Mishne, G. 2008. Finding high-quality content in social media, in: *International Conference on Web Search and Data Mining WSDM '08*, Palo Alto, USA.

Antunes, F. & Costa, J.P. 2012. Integrating decision support and social networks. *Advances in Human-Computer Interaction 2012*(Article 9).

Aubonnet, T., Henrio, L., Kessal, S., Kulankhina, O., Lemoine, F., Madelaine, E., et al. 2015. Management of service composition based on self-controlled components. *Journal of Internet Services and Applications* 6(15).

Auer, S. R., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A nucleus for a web of open data. Semantic web. *Lecture Notes in Computer Science, 4825*, 722–735.

Baroni, A., Muccini, H., Malavolta, I. & Woods, E. 2014. architecture description leveraging model driven engineering and semantic Wikis, in*: IEEE/IFIP Conference on Software Architecture (WICSA)*, Sydney, NSW.

Behkamal, B., Kahani, M., Bagheri, E., & Jeremic, Z. (2014). A metrics-driven approach for quality assessment of linked open data. *Journal of Theoretical and Applied Electronic Commerce Research, 9*(2), 64–79.

Bertino, E. & Lim, H.-S. 2010. Assuring data trustworthiness—concepts and research challenges, in: W. Jonker, M. Petković (Eds.), *Secure Data Management*. SDM 2010. Lecture Notes in Computer Science 6358, Berlin, Heidelberg.

Bhatia, S., Li, J., Peng, W. & Sun, T. 2013. Monitoring and analyzing customer feedback through social media platforms for identifying and remedying customer problems, in: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Niagara, Canada.

Bizer, C. 2007. Quality-driven information filtering in the context of web-based information systems. Ph.D. Thesis, Berlin.

Bizer, C., & Cyganiak, R. (2009). Quality-driven information filtering using the WIQA policy framework. *Web Semantics: Science, Services and Agents on the World Wide web Archive, 7*(1), 1–10.

Boley, H. & Chang, E. 2007. Digital ecosystems: principles and semantics, in: *Inaugural IEEE International Conference on Digital Ecosystems and Technologies* (DEST 2007), Cairns, Australia.

Boris, O., Auer, S., Cirullies, J., Jürjens, J., Menz, N., Schon, J., et al. 2016. Industrial data space: digital souvereignity over data, technical report Fraunhofer-Gesellschaft, doi:10.13140/RG.2.1.2673.0649.

Bosch, J. 2009. From software product lines to software ecosystems, in: *The 13th International Software Product Line Conference* (SPLC'09), San Francisco, USA.

Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data Science Journal, 14*(2), 1–10.

Castillo, C., Mendoza, M. & Poblete, B. 2011. Information credibility on twitter, in: *The 20th International Conference on World Wide Web*, Hyderabad, India.

Chan, C.M.L. 2013. From open data to open data innovation strategies: Creating E-Services Using Open Government Data, in: *The 46th Hawaii International Conference on System Sciences (HICSS)*, Wailea, USA.

Chen, M., Ebert, D., Hagen, H., Laramee, R. S., Van Liere, R., Ma, K. L., et al. (2009). Data, information, and knowledge in visualization. *Computer Graphics and Applications, 29*, 12–19.

Chesbrough, H. W., & Appleyard, M. M. (2007). Open innovation and strategy. *California Management Review, 50*, 57–76.

Daga, E.; d'Aquin, M.; Motta, E.; Gangemi, A. 2015. A bottom-up approach for licences classification and selection, in: *Proceedings of the International Workshop on Legal Domain and Semantic Web Applications (LeDA-SWAn) Held during the 12th Extended Semantic Web Conference* (ESWC 2015).

Dai, C., Lin, D., Bertino, E. & Kantarcioglu, M. 2008. An approach to evaluate data trustworthiness based on data provenance, in: W. Jonke, M. Petkovic (Eds.), SDM 2008. Lecture Notes on Computer Science 5159.

Dobrica, L., & Niemelä, E. (2002). A survey on software architecture analysis methods. *IEEE Transactions on Software Engineering, 28*(7), 638–653.

European Commission. (2011). *Open data an engine for innovation, growth and transparent governance, COM/ 2011/0*. Brussels: European Commission.

Fabijan, A., Holmström Olsson, H., & Bosch, J. (2015). Customer feedback and data collection techniques in Software R&D: A literature review. *Lecture Notes in Business Information Processing, 210*, 139–153.

Ferrando-LIopis, R., Lopez-Berzosa, D. & Mulligan, C. 2013. Advancing value creation and value capture in data-intensive contexts., in: *IEEE International Conference on Big Data*, Silicon Valley, USA.

García, F., Bertoa, M. F., Calero, C., Vallecillo, A., Ruíz, F., Piattini, M., et al. (2006). Towards a consistent terminology for software measurement. *Information and Software Technology, 48*, 631–644.

Gil, Y., & Artz, D. (2007). Towards content trust of web resources. *Web Semantics: Science, Services and Agents on the World Wide Web, 5*(4), 227–239.

Gorton, I., & Klein, J. (2015). Distribution, data, deployment: software architecture convergence in big data systems. *IEEE Software, 32*(3), 78–85.

Guessi, M., Moreira, D.A., Abdalla, G., Oquendo, F. & Nakagawa, E.Y. 2015. OntolAD: a formal ontology for architectural descriptions, in: *30th ACM Symposium on Applied Computing (ACM/SAC'2015)*, Salamanca, Spain.

Hanssen, G.K. & Dybå, T. 2012. Theoretical foundations of software ecosystems, in: Proceedings of the Forth International Workshop on Software Ecosystems (IWSECO), Cambridge, MA, USA.

Heimstädt, M., Saunderson, F., & Heath, T. (2014a). From toddler to teen: growth of an open data ecosystem. *eJournal of eDemocracy & Open Government (JeDEM), 6*(2), 123–135.

Heimstädt, M., Saunderson, F. & Heath, T. 2014b. Conceptualizing open data ecosystems: a timeline analysis of open data development in the UK, in: *Proceedings of the International Conference for E-Democracy and Open Government* (CeDEM2014), Krems, Austria.

HM Government Cabinet Office 2012. *Open data white paper: unleashing the potential*, Retrieved, London, UK.

Iansiti, M. & Levien, R. 2004. Creating value in your business ecosystem. *Harvard Business Review* 2004, 68–78.

Immonen, A., & Niemelä, E. (2008). Survey of reliability and availability prediction methods from the viewpoint of software architecture. *Software and Systems Modeling, 7*(1), 49–65.

Immonen, A., Palviainen, M., & Ovaska, E. (2013). Towards open data based business: survey on usage of open data in digital services. *International Journal of Research in Business and Technology, 4*(1), 286–295. doi:10.0001/ijrbt.v4i1.197.

Immonen, A., Palviainen, M., & Ovaska, E. (2014). Requirements of an open data based business ecosystem. *IEEE Access, 2*, 88–103. doi:10.1109/ACCESS.2014.2302872.

Immonen, A., Paakkonen, P., & Ovaska, E. (2015a). Evaluating the quality of social media data in big data architecture. *IEEE Access, 3*, 2028–2043.

Immonen, A., Ovaska, E., Kalaoja, J., & Pakkala, D. (2015b). A service requirements engineering method for a digital service ecosystem. *Service Oriented Computing and Applications, 10*(2), 151–172.

ISO. (2008). *ISO/IEC 25012—software engineering—software product quality requirements and evaluation (SQuaRE)—data quality model*. Geneva: International Organization for Standardization.

ISO/IEC. (2001). *ISO/IEC 9126–1: Software engineering—software product quality—part 1 : quality model*. Geneva: International Organization for Standardization.

ISO/IEC. (2003). *ISO/IEC TR 9126-2: software engineering—software product quality—part 2 : External metrics*. Geneva: International Organization for Standardization.

Jansen, S. & Cusumano, M. 2012. Defining software ecosystems: a survey of software platforms and business network governance, in*: The 4th International Workshop on Software Ecosystems*, Cambridge, USA.

Kantorovitch, J. & Niemelä, E. 2008. Service description ontologies, in: Mehdi Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology*.

Kazman, R., Klein, M. & Clement, P. 2000. ATAM: method for architecture evaluation, *The 4th IEEE International Conference on Engineering of Complex Computer Systems TECHNICAL*, Carnegie Mellon University, Monterey, USA.

Kett, H., Voigt, K., Scheithauer, G. & Cardoso, J. 2008. Service engineering in business ecosystems, in: *Proceedings of the XVIII International RESER Conference*, Stuttgart, Germany.

Khriyenko, O. 2012. Collaborative service ecosystem—step towards the world of ubiquitous services, in: *Proceedings of the IADIS International Conference Collaborative Technologies*, Lisbon, Portugal.

Li, S. & Fan, Y. 2011. Research on the Service-Oriented Business Ecosystem (SOBE), in: *The 3rd International Conference on Advanced Computer Control* (ICACC), Harbin, China.

Madnick, S. E., Wang, R. Y., Lee, Y. W., & Zhu, H. (2009). Overview and framework for data and information quality research. *Journal of Data and Information Quality, 1*(1), 1–22.

Mecella, M., Scannapieco, M., Virgillito, A., Baldoni, R., Catarci, T. & Batini, C. 2002. Managing data quality in cooperative information systems, in: *Proceedings of the Confederated International Conferences DOA, CoopIS and ODBASE*.

Naumann, F. 2002. Quality-driven query answering for integrated information systems, 2002, Berlin Heidelberg, New York.

Naumann, F. & Rolker, C. 2000. Assessment methods for information quality criteria, in: *The 5th International Conference on Information Quality, Boston, USA*.

Niemelä, E., & Immonen, A. (2007). Capturing quality requirements of product family architecture. *Information and Software Technology, 49*(11–12), 1107–1120.

Niemelä, E., Evesti, A. & Savolainen, P. 2008. Modeling quality attribute variability, in: *Proceedings of the 3rd International Conference on Evaluation of Novel Approaches to Software Engineering*, Funchal, Madeira, Portugal.

Niskanen, I., Purhonen, A., Kuusijärvi, J. & Halmetoja, E. 2014. Towards semantic facility data management, in: *INTELLI 2014, The Third International Conference on Intelligent Systems and Applications*. https://www.thinkmind.org/index.php?view=article&articleid=intelli_2014_4_50_70105 (accessed November 9, 2016).

Nurse, J.R.C., Rahman, S.S., Creese, S., Goldsmith, M. & Lamberts, K. 2011. Information quality and trustworthiness: a topical state-of-the-art review, in: *International Conference on Computer Applications and Network Security* (ICCANS), Male, The Maldives.

Nurse, J.R.C., Agrafiotis, I., Creese, S., Goldsmith, M. & Lamberts, K. 2013. Building confidence in information −trustworthiness metrics for decision support, in: *The 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (IEEE TrustCom−13)., Melbourne, Australia.

Ovaska, E., & Kuusijärvi, J. (2014). Piecemeal development of intelligent smart space applications. *IEEE Access, 2*, 199–214.

Ovaska, E., Evesti, A., Henttonen, K., Palviainen, M., & Aho, P. (2010). Knowledge based quality-driven architecture design and evaluation. *Information and Software Technologies, 52*(6), 577–601.

Ovaska, E., Salmon Cinotti, T. & Toninelli, A. 2012. The design principles and practices of interoperable smart spaces, in: L. Xiaodong, L. Yang (Eds.), *Advanced Design Approaches to Emerging Software Systems: Principles, Methodologies and Tools*.

Pantsar-Syväniemi, S., Kuusijärvi, J. & Ovaska, E. 2011. Supporting situation-awareness in smart spaces, in: *Grid and Pervasive Computing Workshops*, Volume 7096 of the Series Lecture Notes in Computer Science, Oulu, Finland.

Pantsar-Syväniemi, S., Purhonen, A., Ovaska, E., Kuusijärvi, J., & Evesti, A. (2012). Situation-based and self-adaptive applications for the smart environment. *Journal of Ambient Intelligence and Smart Environments, 4*(6), 491–516.

Pipino, L., Lee, Y., & Wang, R. (2002). Data quality assessment. *Communications of the ACM, 45*, 211–218.

Pipino, L., Wang, R., Kopcso, D. & Rybold, W. 2005. *Developing measurement scales for data-quality dimensions*, New York.

Poikola, A., Kola, P., & Hintikka, K. A. (2011). *Public data—an introduction to opening information resources, Ministry of Transport and Communications*. Helsinki: http://www.scribd.com/doc/57392397/Public-Data.

Pollock, R. 2011. Building the (open) data ecosystem. Open Knowledge Foundation Blog, Retrieved from http://blog.okfn.org/2011/03/31/building-the-open-data-ecosystem/on July 10, 2013.

Rafique, I., Lew, P., Qanber Abbasi, M., & Li, Z. (2012). Information quality evaluation framework: extending ISO 25012 data quality model. *World Academy of Science, Engineering and Technology, 65*, 523–528.

Rahman, S.S., Creese, S. & Goldsmith, M. 2011. Accepting information with a pinch of salt: handling untrusted information sources, in: *Security and Trust Management*, Lecture Notes in Computer Science Volume 7170.

Ramaswamy, L., Lawson, V. & Gogineni, S. V 2013. Towards a quality-centric big data architecture for federated sensor services, in: *IEEE International Congress on Big Data*, Santa Clara, CA.

Ruokolainen, T. 2013. A model-driven approach to service ecosystem engineering (PhD Thesis), University of Helsinki, Department of Computer Science, Helsinki, Finland.

Ruokolainen, T. & Kutvonen, L. 2009. Managing interoperability knowledge in open service ecosystems, in: *The 13th Enterprise Distributed Object Computing Conference Workshops*, Auckland, New Zealand.

Ruokolainen, T., Ruohomaa, S. & Kutvonen, L. 2011. Solving service ecosystem governance, in: *IEEE 15th International Enterprise Distributed Object Computing Conference Workshops*, Helsinki, Finland.

Sande, M. Vander, Dimou, A., Colpaert, P., Mannens, E. & Van de Walle, R. 2013. Linked data as enabler for open data ecosystems, in: *Proceedings of the W3C Workshop on Open Data on the Web*, London, UK.

Stathel, S., Finzen, J., Riedl, C. & May, N. 2008. Service innovation in business value networks, in: *The 18th International RESER Conference*, Stuttgart, Germany.
W3C 2007. Web services policy 1.5—framework (W3C Recommendation).
Wand, Y., & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM, 39*(11), 86–95.
Wang, R., & Strong, D. (1996). Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems, 12*(4), 5–33.
Wiesner, S., Peruzzini, M., Doumeingts, G. & Thoben, K.D. 2012. Requirements engineering for servitization in Manufacturing Service Ecosystems (MSEE), in: *Conference on Industrial Product Service Systems* (CIRP IPS2 2012), Tokyo, Japan.
Zhang, J. & Fan, Y. 2010. Current state and research trends on business ecosystem, in: *IEEE International Conference on Service-Oriented Computing and Applications* (SOCA), Perth, Australia.
Zhou, J., Ovaska, E., Evesti, A., & Immonen, A. (2011). OntoArch Reliability-aware Software architecture design and experience. In A. Dogru & V. Bicer (Eds.), *Modern software engineering concepts and practices: advanced approaches*. New York: USA.

Anne Immonen received her MSc degree in 2002 from the University of Oulu. Since 2002, she has been working as a research scientist at VTT Technical Research Centre of Finland, and the last 2 years as a senior scientist. Her main research topics include quality-driven service engineering, digital service ecosystems, and data quality evaluation. Her current areas of interest are quality in digital service ecosystems, quality of data, and data certification for the data-based services. She is currently finalizing her doctoral dissertation in the University of Oulu.

Eila Ovaska received a PhD degree from the University of Oulu in 2000. Prior to 2000, she was a software engineer, a senior research scientist, and leader with the Software Architectures Group at the VTT Technical Research Centre of Finland. Since 2001, she has been a research professor with VTT and an adjunct professor with the University of Oulu. Her current areas of interest are self-adaptive service architectures and knowledge-oriented service engineering in digital service ecosystems. She has acted as a workshop and conference organizer and as a reviewer for scientific journals and conferences. She has co-authored over 150 scientific publications.

Tuomas Paaso (M.Sc. EE) is a research scientist at VTT Technical Research Centre of Finland. He joined VTT in 2005 and has since worked on various research topics concentrating on service architectures, metadata management, and service composition. His current research is focused on service composition and service data models on various domains.

Series title and number

VTT Science 159

Title	**Quality in open data based digital service ecosystems**
Author(s)	Anne Immonen
Abstract	To a growing extent, the software systems of today are provided as digital services distributed across networks, dynamically fulfilling the complex demands of consumers. As people have access to the Internet almost everywhere with the help of the mobile devices, such digital services are expected to be available when requested, and to provide services reliably and without any interruptions. Recently, the use of freely available data on the Internet has increased continuously in the context of digital services. This kind of open data has been identified as providing several benefits to service providers, such as new ideas, services, data-based contents, and confirmation in business decision making. Digital service engineering itself is evolving, and is shifting from isolated development environments towards open innovation and co-development environments, called ecosystems. Digital service ecosystems enable service providers to strengthen their position by cooperating, while still being able to act independently. The ecosystem supports the business models of its actors, also enabling the utilisation of existing ecosystem assets, such as knowledge and services. This dissertation concentrates on the quality of digital service, with an emphasis on open data in ecosystem-based service engineering. The contribution of this research is a concept of an open data based digital service ecosystem, which provides the assets for service providers to design the quality of services and to ensure the quality of open data. These assets include the service engineering model that enables quality-driven service co-innovation and co-development among ecosystem members, the knowledge that can be utilised in digital service engineering, and the enabling environment with knowledge management models and support services for acting in the ecosystem. Additionally, the ecosystem provides support for defining an open business model, for evaluating the quality of open data, and for communication between digital service providers and open data providers. The ecosystem concept is generic, and can be adapted to different application domains; the domain model used together with generic knowledge management models adapts the service engineering and digital services, for example, to the healthcare, energy or traffic domains. The developed concept has been validated incrementally in several application domains.
ISBN, ISSN, URN	ISBN 978-951-38-8557-1 (Soft back ed.) ISBN 978-951-38-8556-4 (URL: http://www.vttresearch.com/impact/publications) ISSN-L 2242-119X ISSN 2242-119X (Print) ISSN 2242-1203 (Online) http://urn.fi/URN:ISBN:978-951-38-8556-4
Date	September 2017
Language	English, Finnish abstract
Pages	102 p. + app. 119 p.
Name of the project	
Commissioned by	
Keywords	Open data, quality, digital service, service ecosystem
Publisher	VTT Technical Research Centre of Finland Ltd P.O. Box 1000, FI-02044 VTT, Finland, Tel. 020 722 111

Julkaisun sarja ja numero

VTT Science 159

Nimeke	**Laatu avoimeen tietoon pohjautuvassa digitaalisessa palveluekosysteemissä**
Tekijä(t)	Anne Immonen
Tiivistelmä	Yhä suurempi osa nykyisistä ohjelmistoista tarjotaan käyttäjille digitaalisina palveluina. Digitaaliset palvelut ovat tyypillisesti tietoverkkoihin hajautettuja palveluja, jotka vastaavat dynaamisesti palvelunkäyttäjien monimutkaisiin ja jatkuvasti muuttuviin vaatimuksiin. Koska ihmisillä on nykyisin pääsy internetiin kaikkialta, erityisesti mobiililaitteiden avulla, he olettavat näiden palvelujen olevan aina saatavilla sekä toimivan luotettavasti, ilman keskeytyksiä. Palveluntarjoajien kiinnostus avoimeen tietoon on viime aikoina lisääntynyt huomattavasti, ja avoimen tietoon perustuvia digitaalisia palveluja on alkanut ilmestyä markkinoille. Avoimen tiedon on huomattu tarjoavan paljon hyötyjä palveluntarjoajille, kuten uusia ideoita, palveluja ja dataan pohjautuvaa sisältöä, sekä vahvistusta ja tukea yrityksen päätöksentekoon. Digitaalinen palvelunkehitys itsessään on siirtymässä kohti avoimia innovaatio- ja yhteiskehitysympäristöjä, joita kutsutaan ekosysteemeiksi. Ekosysteemi tukee toimijoidensa liiketoimintaa ja tarjoaa myös tukea, kuten olemassa olevaa tietämystä ja tukipalveluja, joita eri toimijat voivat hyödyntää omassa toiminnassaan. Tämä väitöskirja keskittyy digitaalisten palvelujen laatuun avointa tietoa hyödyntävässä digitaalisessa palveluekosysteemissä. Tutkimuksen pääkontribuutio on avoimeen tietoon perustuvien digitaalisten palvelujen ekosysteemikonsepti, joka tarjoaa tarvittavan tietämyksen ja aputoiminnot, joiden avulla digitaalisten palvelujen tarjoaja voi saavuttaa laatuvaatimukset ja varmistua myös palvelussa käyttämänsä avoimen tiedon laadusta. Konsepti sisältää laatukeskeisen palvelunkehitysmallin, joka mahdollistaa palvelun innovoinnin ja kehityksen yhdessä muiden ekosysteemin toimijoiden kanssa. Konsepti tarjoaa myös tietämyksen, jota voidaan hyödyntää palvelunkehityksessä, ja ympäristön, joka tarjoaa tietämysmallit ja tukipalvelut ja mahdollistaa niiden hyödyntämisen. Lisäksi ekosysteemi tukee siirtymistä avoimeen liiketoimintamalliin, tarjoaa tukea avoimen tiedon laadunvarmistukseen sekä mahdollistaa kommunikoinnin eri ekosysteemin toimijoiden välillä. Kehitetty konsepti on yleinen ja mukautettavissa eri sovellusalueille. Digitaalisten palvelujen kehitys voidaan mukauttaa esimerkiksi terveydenhoidon, energian tai liikenteen sovellusalueelle käyttämällä sovellusaluekohtaista mallia yhdessä yleisen tietämysmallin kanssa. Kehitetty ekosysteemikonsepti on varmennettu asteittain toteuttamalla osittaisratkaisuja eri sovellusalueiden ongelmiin.
ISBN, ISSN, URN	ISBN 978-951-38-8557-1 (nid.) ISBN 978-951-38-8556-4 (URL: http://www.vtt.fi/julkaisut) ISSN-L 2242-119X ISSN 2242-119X (Painettu) ISSN 2242-1203 (Verkkojulkaisu) http://urn.fi/URN:ISBN:978-951-38-8556-4
Julkaisuaika	Syyskuu 2017
Kieli	Englanti, suomenkielinen tiivistelmä
Sivumäärä	102 s. + liitt. 119 s.
Projektin nimi	
Rahoittajat	
Avainsanat	Avoin tieto, laatu, digitaalinen palveu, palveluekosysteemi
Julkaisija	Teknologian tutkimuskeskus VTT Oy PL 1000, 02044 VTT, puh. 020 722 111

Quality in open data based digital service ecosystem

New quality challenges in digital services have recently arisen, caused by the new innovation and co-development environments (called ecosystems), the growing number of customer-controlled services available dynamically online, and the use of open data in digital services. This dissertation investigates open data based digital service engineering in the ecosystem context, concentrating on the quality of services.

The contribution of this dissertation is an evolvable open data based digital service ecosystem (EODE) concept that provides a new cooperating environment for the actors of digital service ecosystems and open data based business ecosystems, supporting the businesses of all its actors. EODE specifies what kinds of actions are required to capture quality in the service design and to ensure the quality of the open data utilised in digital services. It also provides the infrastructure with the knowledge management models and supporting services to implement these actions. EODE is generic, and can be adapted to different application domains with domain specific models.

ISBN 978-951-38-8557-1 (Soft back ed.)
ISBN 978-951-38-8556-4 (URL: http://www.vttresearch.com/impact/publications)
ISSN-L 2242-119X
ISSN 2242-119X (Print)
ISSN 2242-1203 (Online)
http://urn.fi/URN:ISBN:978-951-38-8556-4