Veikko Seppänen

Competence change in contract R&D

Analysis of project nets

TECHNICAL RESEARCH CENTRE OF FINLAND ESPOO 2000

VTT PUBLICATIONS 418

Competence change in contract R&D

Analysis of project nets

Veikko Seppänen

VTT Electronics

Academic dissertation to be presented with the assent of the Faculty of Economics and Industrial Management, University of Oulu, for public discussion in Auditorium L10, on November 10th, 2000, at 12 noon.

TECHNICAL RESEARCH CENTRE OF FINLAND
ESPOO 2000

ISBN 951–38–5577–5 (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–5578–3 (URL: http://www.inf.vtt.fi/pdf/)
ISSN 1455–0849 (URL: http://www.inf.vtt.fi/pdf/)

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (90) 4561, telekopio 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (90) 4561, telefax 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, P.O. Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 0 4561, telefax + 358 0 456 4374

VTT Elektroniikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Technical editing Leena Ukskoski

Otamedia Oy, Espoo 2000

Seppänen, Veikko. Competence change in contract R&D. Analysis of project nets. Espoo 2000. Technical Research Centre of Finland, VTT Publications 418. 226 p. + app. 29 p.

Keywords core competence, customer relationships, R&D management, industrial networks, software engineering, fault diagnosis, code generation, electronics industry

ABSTRACT

This research addresses the problem of what to consider as competence in research and development (R&D) and how does it change in contractual projects. Two cases are investigated, so called fault diagnosis systems and code generation methods and tools. They involve the Technical Research Centre of Finland (VTT) as a supplier and several companies as customers. Projects established by these parties from 1985 to 1998 have been analysed, to study how the supplier's competence evolves. The research is based on the analysis of several thousand pages of documents and interviews of some fifty persons. The change of competence is evaluated and explained within project-based relationships called project nets. Differing objectives and goals of the interacting parties are found to greatly affect project nets, and thereby the evolution of the supplier’s competence.

In the code generation case, the focal VTT managers aimed at creating a versatile portfolio of fully contractual projects, for machine automation applications in particular. The approach did not work as intended. The code generation researchers produced design methods and tools that were, in the end, mostly utilised inside VTT – which neither benefited the contractual R&D business of VTT, nor resulted in explicit core competence. In the fault diagnosis case contractual projects were, however, used to create what can be considered as core competence of VTT as an R&D supplier.

Still, even the code generation competence survived through many years and conflicting viewpoints to form a basis of business – not for VTT, but for some of the original inventors. This indicates both the key role of individuals in R&D and the need for strategic management of evolving competence. Development of functional capabilities to solve problems in certain applications is found in this research to be essential for the building of core competence of an R&D supplier. Flexible use of generic engineering techniques and implementation technologies is also important. However, this depends on how skilled customers are in this regard.

The final part of the dissertation is devoted to the question of how to manage competence, based on the empirical insights of the research. The goal is to help pave the way for strategic relationship-based competence management in contractual R&D.

PREFACE

The roots of this research can be traced back to 1993, when I started my executive MBA course at the University of Oulu. Based on these studies, I took an interest in the "improvement" of the R&D process. As a graduation present I received the book by Hamel and Prahalad on core competence. The book was one of the stimuli to this research. Having been a software engineering researcher for fifteen years, I could not help wondering what competence actually meant in practice, in my daily working environment. I saw an opportunity, as a business science researcher, to investigate and express in explicit terms some elements of competence and to analyse its evolution in the context of project-based relationships. This is what I have been doing during the past three years.

Professor Kimmo Alajoutsijärvi, the advisor for my research, comes not from the engineering but the marketing research community. He made it clear from the beginning that my research should not result in yet another dissertation on computer technology, but in a thesis on business science. It is no exaggeration to say that this research would not have been possible without his help. As an expert in industrial marketing and a researcher who was thoroughly familiar with longitudinal case studies, he was guiding me to the right direction. I must say that even if it has not always been easier, at least it has been much more fun to follow this direction, compared to the conventional engineering research process.

Professors Päivi Eriksson, Pekka Kess, Kristian Möller and Henrikki Tikkanen helped me in a most practical way during my research. I am indebted to them for their time and patience, and now understand how frustrating it must have been to try to teach me some of the very basic issues of business research. I am especially grateful for Henrikki's efforts in this regard. The same holds for the reviewers of the manuscript of the thesis, Professor Håkan Håkansson and Dr. Carl-Johan Rosenbröijer – their comments improved this final version of the dissertation remarkably.

Several people at VTT and elsewhere have helped me to carry out the two case studies included in the research. Please accept my hearty thanks and sincere apologies for not mentioning you here in order to guarantee a minimum of privacy for your past endeavours. I have tried to present your points of view in this dissertation as clearly and honestly as possible. The responsibility for any mistakes in the conclusions drawn from your interviews and comments is solely mine. At this point, I would also like to thank Dr. Kari Leppälä, Dr. Keijo Räsänen and Dr. Ahti Salo, who helped me in formulating my research plan and dealing with the two cases.

Professor Jorma Lammasniemi made it possible for me to carry out this research alongside my daily duties at VTT Electronics. I am very grateful for his support. Ms. Anu Angeria, Mrs. Eija Posio, Mrs. Sari Mäläskä, Mrs. Eija Hyvönen and Ms. Leena Ukskoski helped me to organise the research.

Dr. Matti Kurki, Dr. Markku Oivo, Mr. Ari Okkonen, Mr. Jarmo Kalaoja and Mr. Harri Perunka were more useful for my work than they perhaps themselves know. First of all, even though they very well could have done, they did not denigrate this research. I especially wish to thank Mr. Jukka Karjalainen for the opportunity to use his personal R&D diaries covering many years. If only all of us would be as well organised in terms of personal remarks as Jukka is, case study researchers would have an easy life. Not to mention his openness and willingness to make his thoughts available for a curious investigator.

I would like to apologise for not comprehending the lesson Dr. Risto Suitiala tried to teach me in the early nineties on the importance of understanding why and how people behave, rather than trying to “improve” their work. I have now taken a long course in understanding, a course which Risto obviously passed with an excellent grade years ago.

I have analysed in the final part of the dissertation managerial issues, based in part on a strategic planning process carried out with the help of Mr. Pertti Pekkala and my colleagues at VTT Electronics. This was certainly useful to ground some of the basic ideas of my research to the everyday management practices of a contract research organisation. I am grateful to Pertti, Seija, Tua, Jarkko, Hannu Honka and Hannu Rytilä, Petteri, Johan, Tapio, Janne, Jorma Taramaa and Jorma Hintikka, and Markku for their interest in coping with managerial changes.

I would also like to thank the fellow graduate students on marketing, from whom I learned a lot, and with whom it was always a pleasure to associate. Discussions with Minna Leminaho, Tuula Mittilä, Päivi Seppälä and Jaana Tähtinen, among others, helped me to realise that it really is possible not only to survive graduate studies, but also to have some fun every now and then.

The Foundation for Economic Education and the Foundation of The Association of Electronics Engineers in Finland provided personal grants for me to carry out the research published in this dissertation, which is most gratefully acknowledged. Thanks are also due to Mr. Seppo Keränen and Mr. David Cox, who proof-read the texts of this dissertation – several times and in quite many pieces.

I do not dare to promise that this is the last dissertation that I will ever write, but for a while at least, I am going to heavily focus on my relationships with Päivi, Pekka, Katri and Jaakko instead of further studying industrial networks during my spare time. I am afraid that my competence in family matters needs improvement – in practice, not in theory.

Oulu, July 20, 2000

Veikko Seppänen

CONTENTS

ASTRACT 3

PREFACE 4

SYMBOLS AND ABBREVIATIONS 10

PART I: SETTING THE SCENE........ 11

1 INTRODUCTION 11
1.1 Background 11
1.2 Context – contractual R&D projects 13
1.2.1 Discussion – research and praxis of R&D 16
1.3 Research problem 20
1.4 Dissertation structure 22

2 RESEARCH DESIGN 23
2.1 Overview of the research process 24
2.2 Case study method 34

3 ANALYSIS OF RELATED WORK 41
3.1 Competence studies 41
3.1.1 Resource-based views to competence 43
3.1.2 Characteristics of core competence 49
3.1.3 Core products, platforms and product families 51
3.2 Research into industrial networks 55
3.3 Summary 62

PART II: SUBSTANCE OF COMPETENCE 63

4 INITIAL FRAMEWORK 63
4.1 Substance layer of the framework 63
4.2 Dynamic layer of the framework 67
4.2.1 Changes among relationships 67
4.2.2 Changes within relationships 68
4.2.3 Managing of competence changes 70

5 THE FAULT DIAGNOSIS CASE 74
5.1 The fault diagnosis substance 74
5.2 Changes in project relationships 76
5.3 Evolution of the fault diagnosis competence 80
5.4 Preliminary empirical findings 85
5.5 Initial theoretical contributions 89
5.6 First managerial implications 90
5.7 Problem of contextual analysis 93

PART III: COMPETENCE EVOLUTION IN FOCAL NETS 94

6 REVISED FRAMEWORK 94
6.1 Process-based viewpoint to contract R&D 94
6.1.1 Business processes and networks 95
6.1.2 Process-based extensions of the framework 98
6.1.3 Discussion 99
6.2 Knowledge creation versus competence 100
6.3 Explaining the logic of action 103
6.4 Summary 107

7 THE CODE GENERATION CASE 109
7.1 Introduction 109
7.2 MACRO FORCES OF THE OUTER CONTEXT 114
7.2.1 Economy – Finnish electronics industry 1985–1998 116
7.2.2 Technology – embedded software engineering 122
7.3 GROUPS OF ACTORS INVOLVED IN PROJECT NETS 125
7.3.1 Objectives, goals and logic of action 125
7.3.2 Planned versus realised project nets 131
7.3.2.1 Mutuality – forms of interaction 138
7.3.2.2 Mutuality – scanning and diffusion 141
7.4 CONTENT OF LOGIC OF ACTION – CODE GENERATION PROCESSES 146
7.4.1 Activities on resources – assuring of capability 150
7.4.2 Activities on resources – balancing of particularity 153
7.5 Evolution and valuation of the competence 156
7.5.1 Competence during the Speco period 156
7.5.2 Competence during the Sokrates period 158
7.5.3 Competence during the Reagenix period 165

8 ANALYSIS AND IMPLICATIONS OF THE RESULTS 169
8.1 Analysis of the competence evolution 169
8.2 Cross-case analysis 173
8.3 Lessons learned 179
8.3.1 Insights for research 179
8.3.2 Theoretical findings 181
8.3.3 Managerial implications 182

PART IV: STRATEGIC COMPETENCE MANAGEMENT 186

9 MANAGERIAL VIEW TO COMPETENCE 186
9.1 Introduction 187
9.2 Analysis of managerial implications 189
9.3 Relationship-based competence management 193
9.3.1 Portfolio design for maintaining customer focus 194
9.3.2 Concurrent R&D for core platforms 201
9.4 Discussion of the model 210

10 CONCLUSIONS.. 213

REFERENCES... 216

APPENDICES
  Appendix 1. The fault diagnosis story
  Appendix 2. The code generation stories

SYMBOLS AND ABBREVIATIONS

AAAI	American Association of Artificial Intelligence
AI	Artificial intelligence
ARA	Activity-resource-actor model
ASIC	Application-specific integrated circuit
AUT	VTT Automation
C	A programming language
CASE	Computer-aided software engineering
CBR	Case-based reasoning
CERN	European Laboratory for Particle Physics
CMM	Capability maturity model
COTS	Commercial off-the-shelf (software)
CRO	Contract research organisation
CUTE	C software unit testing tool
GSM	Mobile telephone standard
DSP	Digital signal processing
EDA	Electronic design automation
ELE	VTT Electronics
ELI	VTT Food technology laboratory
EU	European Union
FIM	Finnish mark
IJCAI	International Joint Conference on Artificial Intelligence
IMP	Industrial Marketing and Purchasing
IMS	Intelligent manufacturing systems
IPR	Intellectual property rights
IR	Inter-organisational relationship
IT	Information technology
KE	Knowledge engineering
KEE	Knowledge Engineering Environment
KTM	Ministry for Trade and Industry
OO	Object oriented
PC	Personal computer
QFD	Quality function deployment
R&D	Research and development
SA/SD	Structured analysis and design
SDL	Structured design language
SPC	Statistical process control
STUK	Finnish Centre for Nuclear Safety
TKK	Technical University of Helsinki
TKO	VTT Computer technology laboratory
TQM	Total quality management
TTCN	Tree and tabular combined notation
UIF	User interface
VLSI	Very large scale integration of electronic circuits
VR	Virtual reality
VTT	Technical Research Centre of Finland
WWW	World Wide Web

PART I: SETTING THE SCENE

This dissertation is divided into four parts. *The first part* introduces the context of the study, presents the research problem and discusses the research design used in solving the problem. An overview of the research process is also provided. The subsequent parts follow the process that I went through during the past three years. This certainly results in some overlapping, but since the process was a true learning experience, I wanted to make this clear in the dissertation. *The second part* of the dissertation focuses on what can be understood as competence in contractual engineering research and development (R&D). I have approached the issue by analysing one of the two cases of my research. *The third part* addresses the change of the competence of an R&D supplier in project-based relationships. The study exploits the analysis of the second case. The analysis is largely based on the projects carried out by the supplier to create and make use of resources in the context of R&D tasks, which is my view to competence. Both individuals and organisations are addressed as actors in the projects. I have gathered and interpreted their views to the competence of the focal supplier, as it has evolved in projects. Since competence is a means for business, not an end, *the fourth part* of the dissertation focuses on strategic management of competence for the needs of contractual R&D.

1 INTRODUCTION

The first chapter introduces the background of the research, and later discusses both the research problem and the context of the study.

1.1 BACKGROUND

This study was carried out as a piece of business research, although it also focuses on organisational and some strategic management issues. The reason for this is the fact that competence changes in inter-organisational relationships involve, in practice, both organisational and managerial matters. One of the main motivations for this research is that professional services is an important but rather poorly understood area of business (cf. [Halinen 1994]). *Contractual R&D services* are a special domain in the borderline of science, technology and business, where science-based innovations must be produced while making business at the same time. In professional *engineering R&D services* technology development forms yet another dimension to tackle. This research demonstrates the various views different people may have on the same R&D subject: one participant sees it as a scientific problem, another as the development of a new technology, and a third one as a business case. In large corporations different views may be handled by different organisational units, but in small supplier organisations the views must be united. This is a practical managerial problem that can be addressed only after understanding the essential nature of competence and how does it change over time.

Several theoretical and empirical studies have shown that collaboration with external partners provides a valuable means for organisations to foster innovation, and to improve the use of their resources. These studies have identified many kinds of potential collaboration partners, such as customers, suppliers, and competitors, capable of contributing to the development and deployment of the resources of the focal organisation. To name a few, I would like to mention Håkansson [1987], Hamel and Prahalad [1994], Gemunden and Ritter [1996], Eriksson and Keso [1997], and [Alajoutsijärvi and Tikkanen 1999a,b].

Some people, similarly as I in this research, have investigated the creation and evolution of a company's resources through R&D, for example Gallon et al. [1995]. Other studies have concerned the customer relationships in the services sector, e.g. Halinen [1994]. However, only a few authors have treated the subject of professional R&D service organisations from the viewpoint of external relationships. Focusing on such an organisation, I address competence changes in project-based relationships. According to Reger and Schmoch [1996], *research and experimental development* is creative work being carried out systematically to increase knowledge to build and take in use new applications.

In contrast with a majority of the literature concerning competence within organisations, I have investigated how inter-organisational relationships affect competence – and what competence actually means in the context of R&D. I present a review of selected frameworks and concepts for competence and relationships in Chapter 3, to create a basis for addressing them together. When seen from the managerial point of view, the two areas are clearly interwoven (cf. [Alajoutsijärvi and Tikkanen 1999a]): development of relationships involves the identification and usage of competence, and vice versa. Since I am interested in R&D management as a profession, I will address this issue in the fourth part of the dissertation.

In this study I will concentrate on the issue of competence developing over time in the project-based relationships of one focal organisation, i.e. in focal nets that are a part of some greater industrial network structure. Leppälä [1995] (p. 142) illustrates this by stating that a contract R&D project is usually conducted by a "mixed contractor-client team", focusing on "results and costs" of the project by controlling progress, quality and the contract. At the supplier side, projects are associated with more general "R&D activity and service business", at the customer side with "product projects" and "product centered business activities".

The main function of focal nets, as I have utilised them in this research, is to capture phenomena having affected the competence of the R&D supplier. I have defined the boundaries of focal nets based on projects since they reflect the actual operational form of inter-organisational interaction of the supplier addressed in my study. I present a model, called the competence evolution framework, to make explicit and evaluate changing competence in project-based focal nets.

Focal nets are modelled in this research by using the four key concepts of *actors, relationships, resources* and *activities* (cf. [Håkansson and Snehota 1995]). Actor structures, bonds, and webs form a governance structure, where competence evolution is being analysed. Even more significantly, the concept of activities performed to create and make use of resources provides for an immediate link between the resource-based view to strategic management of firms (cf. [Sanchez at al. 1996a, Wernerfelt 1984]) and the so called network or interaction model of industrial relationships (cf. [Möller 1994, Möller and Wilson 1995a, Håkansson and Snehota 1995]).

Analysis of certain customer relationships where many different types of competence can develop, has been a more typical focus in industrial network studies (cf. [Rosenbröijer 1998, Easton and Araujo 1996]). A reason for me focusing on certain competence and many relationships is the fact that a contract research organisation seldom can afford to focus on a single party when developing and delivering its competence.

I wish to close this introduction by pointing out that trying to fit my work into a specific discourse of business science was not a simple task. The reason is not only the contractual R&D services having not been extensively studied, but also the fact that many of the basic concepts related to competence, such as resources, are somewhat overlapping and fuzzy. Laamanen [1997], for example, claims that "there seems to be some conceptual confusion due to the newness of the research area" of competence studies. I will discuss the positioning of my work with respect to the related research in Chapter 3.

1.2 CONTEXT – CONTRACTUAL R&D PROJECTS

This research concerns VTT, the Technical Research Centre of Finland (www.vtt.fi), which is one of the largest Scandinavian contract research organisations, and has a history of decades of engineering research and development [Michelsen 1993, Tiittula 1994]. I focus on VTT Electronics or ELE (from 1994 onwards), and one of its predecessors, VTT Computer technology laboratory or TKO (1983–1993). I have investigated the change of certain competence in the institutes from the mid-eighties to the present day. This kind of an approach has been recommended in some recent technology management studies, for example in [Reger and Schmoch 1996].

VTT is a fully government-owned multi-faceted organisation, offering professional engineering research and development services mostly to Finnish industry (cf. [Tiittula 1994, Leppälä 1995]). The services of the VTT institutes on which I focus cover a wide range of electronic technologies from microelectronics to computer software. The skills mastered by the institutes correspondingly include a variety of engineering techniques. In addition to electronics companies, they have co-operated with companies that use electronics mainly as a supporting technology. In practice, this allows the influence of end-users or inexperienced customers through certain relationships, as opposed to collaboration with technology experts of the leading electronics companies.

This influence can be seen in the two cases I have studied, industrial fault diagnosis systems and code generation methods and tools. The former are used in different kinds of electronic products to monitor and diagnose the condition of the product, and to help error recovery. The latter involve techniques intended to automate industrial software development. The fault diagnosis applications developed by TKO and ELE range from process control and machine automation to electronics assembly lines, moving vehicles, electronic instruments, and telecommunication networks. In other words, VTT possesses competence in fault diagnosis that has been exploited by many industries. The rights to the code generation techniques owned by VTT were sold to a small company in 1998, after several years of declining activities. Although VTT apparently failed to maintain its code generation competence, the small company that bought the rights has been quite successful so far. Therefore, the code generation competence did not disappear either, but is now being utilised in another context.

In both cases, one of the main technologies involved is embedded software, i.e. computer programs incorporated in hardware hidden inside electronic products and systems. Software engineering techniques have been used extensively in the development of both fault diagnosis systems and code generation methods and tools. Especially in the former, the so-called knowledge engineering techniques were also utilised. Knowledge engineering can be regarded as a special area of software engineering.

Organisational setting

In this research I focus on rather small groups of actors, project teams and research groups of one organisation. In contrast, I mainly view external actors at the less refined level of whole companies, since I do not analyse the evolution of their competence due to the projects that they contracted from VTT. The Software engineering and Knowledge engineering research groups are the two focal groups that I have studied. Their size vary from ten to twenty people, compared to the over three hundred people employed at present by ELE. Most of the researchers in the two groups are university graduates in electronics and computer engineering.

Almost all of the groups' activities are carried out as projects that form flows of interaction both within the focal organisation and with customers and other external parties. Projects usually involve several people, whose skills are integrated and sometimes overlap, come they from VTT, customer companies, or other collaborators. This situation differs from an inter-organisational relationship, where the supplier is an expert service organisation, while the buyer does not have a similar capability (cf. an advertising agency and its client in [Halinen 1994]). It is also different from a relationship where the supplier provides dedicated technologies in which the buyer has decided to invest, but does not have the capability to develop these himself (cf. a paper machine manufacturer and paper mills in [Alajoutsijärvi 1996]). However, VTT researchers are not usually experts in the customers' business – even most of the customers' representatives involved in joint projects are engineering rather than business experts.

One of the results of the situation is that the customers' own employees may be partners and competitors as far as certain research and development tasks are concerned, although their expertise is often more application-oriented than that of the VTT researchers. However, individual opinions and skills concerning R&D goals and results, as well as competition for internal and contracted resources, may strongly affect the evolution of an inter-organisational relationship. This is one of my main motives when focusing on “low-level” organisational and project groups.

R&D project portfolio

Financed mostly by ELE itself, strategic research projects are targeted to create certain competence. These *“green”* projects include about 20% of the total project portfolio. Public funding bodies, VTT, and industrial partners finance other projects jointly. Their purpose is the further development and preliminary deployment of competence. These *“blue”* projects comprise about 40% of the project portfolio. The rest, the so-called *“red”* projects, are carried out for individual customer companies or sometimes for a consortium of a few companies. Industrial financing of projects increases correspondingly; close to zero percent in green, twenty percent in blue, and a hundred percent in red projects.

This kind of project portfolio is believed to support technology transfer from research to industrial practice, as well as to reduce technical risks involved in developing products based on new technologies, see Figure 1. In terms of the typology presented by Alajoutsijärvi [1996], the portfolio includes red projects intended for “technology-transfer”, i.e. new competence is transferred to current or new customers. Green projects are for “learning”, i.e. new competence is created, often in collaboration with current key customers, while blue projects are for “breakthroughs”, where new competence is created or extended with new and current customers. “Routine” red projects to deliver existing competence to current customers may also be carried out. However, these projects are tried to be avoided.

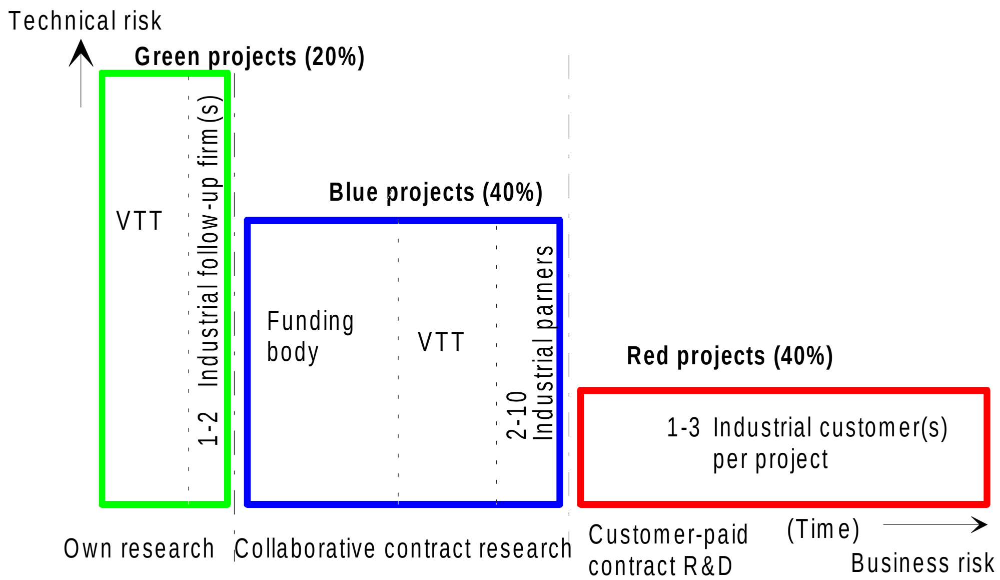

Figure 1. The intended R&D project portfolio.

Technical risks are often high in green projects. On the other hand business risks are low, both for VTT and for possible industrial beneficiaries of the research. These projects are thought to aim at “capability development” from the viewpoint of inter-organisational relationships [Håkansson and Snehota 1995], and blue projects at “strategy development”. In the case of VTT multi-party blue networks of project relationships are most often expected to precede dyadic red customer projects. Of course in practice, the situation is much more complicated than it appears to be in Figure 1, as can be easily seen also from the cases that I have studied in this research.

The relationships are supposed to evolve from loose, interest-group type of networks or dyads associated with green projects, via collaborative multi-party networks sharing costs and benefits in joint blue projects, to fully commercial red customer projects. The latter are most often dyads, possibly involving small networks of two to three customers. The income from red projects should not only cover all the expenses, but also result in a certain yet small profit for VTT, in addition to benefits for customers.

Green and blue projects usually last for two years, and involve less than ten people. Their typical budget is a few million Finnish marks. Tekes, the National Technology Agency of Finland (www.tekes.fi), and EU (the European Union) are the two biggest funding sources for blue projects. The financing scheme covers all project expenses, but may not produce profit for VTT – only its own expenses should be covered. Most red projects are both shorter and smaller in terms of personnel and money involved. However, work is frequently continued on the same subject in several subsequent red projects. The longest red project chains have lasted for three to four years. In the two cases I have studied for this research, the project chains were somewhat shorter on average, from two to three years, and there were certain dormant periods in between the respective projects.

The well-known technology push principle of innovation encourages customers to be involved in certain compartments of the project portfolio (cf. [Lindman 1997]). Another perspective to customer relationships is the separation of key customers from occasional partners, simply on the basis of financial volume of collaboration. The former are taken care of by establishing written or oral co-operation agreements, the latter are usually acquired through distinct project marketing activities.

1.2.1 Discussion – research and praxis of R&D

I will use the definition of R&D by Reger and Schmoch [1996] in this dissertation. Reger and Schmoch consider research and experimental design to “comprise creative work undertaken on a systematic basis in order to increase the stock of knowledge, including knowledge of man, culture, and society, and the use of this stock of knowledge to devise new applications.” It is important that this definition emphasises that R&D involves systematic, albeit creative, work. Moreover, the view that it is necessary to use knowledge gained through R&D for the needs of new applications is essential for such contract research organisations as VTT.

Understanding the essence of R&D – examples of scientific views

[Lindman 1997] is a good example of scientific research of R&D. Lindman conducted a longitudinal study on the development of a family of electronic products, including an in-depth analysis of the perspectives taken and the concepts used to study new product development. He categorises (p. 17) "central research approaches in new product development" as follows:

- resource-based theory of a firm and learning (focusing on development and utilisation of new product capabilities),
- strategy and strategic management (factors of new product performance),
- organisational theories (management approaches towards new products),
- marketing management (market-pull innovation models), new product success/failure models (factors of success and failure), and
- adoption/diffusion theories and technology/innovation management (technology push innovation models).

The competence literature that I will discuss in Chapter 3 has its roots in the resource-based theory. Lindman has also used the resource-based theory in his research. Regarding the roots of this approach, he concludes (pp. 68–83) that the "latest research has indicated that … to stay ahead, firms have to develop new [product] platforms at the same time that they complete products based on the current [product] platform". I share this opinion, and have therefore included a section on core products and platforms in Chapter 3. However, Lindman focuses largely on the performance of new product development, an issue not studied in this research.

[Reger and Schmoch 1996] is a comprehensive study of R&D, carried out for the needs of strategic policy implications. The authors characterise research and development as an example of "modern systems of knowledge production" that is "interdisciplinary, application-oriented, and network-dominated". According to Reger and Schmoch, "the most appropriate way to describe the knowledge transfer between non-industrial institutions and industry is the so-called network approach" – they refer to several existing network models in this regard, and also present their own approach. I will as well make an overview of selected network approaches in Chapter 3, since I have applied the network model to the research.

Direct research contacts, such as contractual projects I have analysed, are according to Reger and Schmoch the most effective means for knowledge transfer. On the other hand, they claim that the emergence of "techno-scientific communities" affects knowledge transfer, because "the channels of communication are largely determined by the scientific character of the examined area". In my opinion this view omits a business perspective to R&D, a perspective inherent in contractual R&D services.

Reger and Schmoch are of the opinion that “basically the relationship between non-industrial research institutions and industry may be described as interaction”, since the relationship is bi-directional. Furthermore, they claim that confidence in an R&D relationship based on a contract may not lead to a balance of interests, but can favour one of the contracting parties. I have investigated the development of the R&D supplier’s core competence, which I consider as one possible way for balancing interests.

However, according to Reger and Schmoch the concept of core competence is too static, since it has “not been sufficiently operationalised, the process of identification within the company as well as the allocation of resources is unclear; further research is necessary”. They claim that core competence thinking can be traced back to the sometimes excessive growth of large corporations and the unsuccessful diversification that resulted: core competence thinking fundamentally contrasts with the strategic business unit concept. In their opinion Leonard-Barton [1992] is “the only author to build a bridge explicitly towards R&D management”, because she addresses competence creation and evolution in the context of R&D projects.

I have also investigated this context, to make explicit and understand competence changes. However, Reger and Schmoch consider that it may “take ten years or more to build up” core competence, and that “without a clear vision of core competence and the strategic architecture of the enterprise, it is not possible to engage meaningfully in co-operation”. In my opinion this view over-emphasises the idea of building competence inside the organisation, as opposed to inter-organisational relationships.

Concerning the managerial implications of R&D studies, Reger and Schmoch claim that “there is no best practice [of R&D], since the use of tools depends on the specific context and situation of the enterprise”. Moreover, “most models for R&D management are based upon observations in large international and multi-divisional enterprises”. I find this view easy to agree with, for even studies of R&D at VTT have addressed either the organisation [Tiittula 1994] or its institutes [Leppälä 1995] as a whole. In this research, I have taken a much narrower view by investigating rather small organisational groups and project teams, two particular competence areas, and less than thirty individual projects.

Reger and Schmoch point out that a practical problem involved in managerial implications in this context is that “R&D management is, like innovation management, a young sub-discipline in business administration”. Considering a contract research institute as a focal supplier in this regard, it is helpful that certain explicit structures and procedures exist intended for managing R&D as a business. An interesting remark related to this is that “only those capable of creating knowledge are capable of absorbing it”, because R&D collaborators “have to solve problems one of the partners is not able to solve alone”. One of my cases indicates the very same: if a customer considers himself as an exploiter or maybe even an end-user of a new technical solution developed by the research institute, great difficulties regarding the use and maintenance of the developed solution may emerge. However, in the other case former exploiters turned successfully into active maintainers and developers of the solutions built by VTT.

Improving R&D services – a marketing practitioner's view

Sipilä [1992] has given several courses at VTT on training researchers and managers to improve contractual R&D services from a marketing perspective. To provide an interesting contrast to the scientific views of R&D discussed above I will finish this section with a brief overview of some of his key ideas. According to Sipilä, service organisations can be classified based on their dependence on experts (high/low), and the form of their capital investment (high/low). The position of VTT in this quadrangle is high dependence on experts and low capital investment.

In Sipilä's opinion, marketing of expert services includes internal and project marketing, choice of customers, use of customer references, and the actual service. This is a rather linear and idealised view. In particular, it neglects both differences in the kinds of project types discussed above, and co-ordination of interrelated supplier and buyer activities in planning for and conducting projects, fundamental activities at VTT.

The competing factors of an expert service organisation are, according to Sipilä, the customer base, reputation, technical and functional quality of the services, productisation and product support, special skills and development capacity, coverage and versatility of the services, price competitiveness and productability, distribution capacity, networking, strategic vision and flexibility. These factors are quite abstract and represent the traditional marketing management perspective (cf. product, price, place alias distribution capacity and promotion alias product support) to customers.

Sipilä considers relationship marketing viable when the following conditions are met: the customer frequently requires services, the supplier has long-term commitment to the market, a few main customers dominate the market, acquiring new customers is very expensive, there are only a few service providers, customers themselves wish to create long-term relationships, quality and reliability are critical competitive factors, and the supplier's service development requires deep customer involvement. I consider the last two issues relevant for VTT, the other being less obvious.

Sipilä proposes market segmentation variables based on customers' industry, type (public, private, etc.), size, types of requirements, economic situation, developmental phase, own expertise, business increase, location, export trade, and status. In this research I have taken a relationship-based view of markets. This comes closer to the project-driven core business of VTT, as has been characterised e.g. by Leppälä [1995].

According to Sipilä, a product strategy in expert services should include the levels of competence areas, knowledge bases, products, and product versions. I have approached this question by studying what can core competence mean in the context of contractual R&D projects, and how does it develop. The idea of explicit core competence in service business, called the "core platform" by Hamel and Prahalad [1994] is not similar to the product strategy proposed by Sipilä, since a core platform is "an intermediate [service] product somewhere between core competence and the end-product" [Hamel and Prahalad 1994].

1.3 RESEARCH PROBLEM

This research started from a process improvement perspective, as I was interested in what I called the “maturity” of R&D as a process. The research then focused on understanding what competence is and how it changes in inter-organisational project relationships. Understanding and explaining both competence and interaction in projects was not at all straightforward. However, looking back over the past years and trying to recapture some of the success factors and pitfalls in certain projects, I could return to the problem of competence management as a somewhat wiser man. One of the best lessons I learnt is that it is overly naive to imagine competence developing solely within one organisation.

The problem I originally planned to address changed considerably after my research had already begun, or maybe it was my view of the problem that changed. The phenomenon I was interested in, i.e. competence in the context of contractual R&D, remained much the same. Initially, I addressed the managerial problem of “How to increase the maturity of R&D” by the means of competence. The problem was then reformulated to a more explorative question of “How customer relationships are affected by competence changes”. Finally, the problem fell into a reversed problem of studying how competence changes in R&D projects:

How does competence evolve in R&D project relationships?

To solve the problem it was necessary to answer the following questions:

1. what does competence mean in the context of contractual research and development (and how to conceptualise both competence and project-based R&D relationships),

2. how does the competence of the focal organisation change along time in R&D projects (and how to make changes explicit),

3. what affects competence evolution in the context of R&D projects (and how to illustrate it), and

4. how to manage competence evolution in practice within R&D projects?

The first question is addressed in the second part of my dissertation, through an analysis of the fault diagnosis case, and using an initial version of the competence evolution framework. To answer the second and third questions, competence changes and their reasons are investigated in the third part of the dissertation, on the basis of the code generation case. The case was analysed by utilising a revised version of the competence evolution framework. The fourth question concerning what I call strategic relationship-based competence management, is addressed in the final part of the dissertation.

Motivation for solving the research problem

The first question is highly relevant both conceptually and managerially. There are no uniform competence theories available (cf. [Sanchez et al. 1996a]), not even any widely shared understanding of what competence means (cf. [Laamanen 1997]). To be able to answer the question, I have combined, extended and applied existing concepts for contract research and development, as will be described in the second and the third part of the dissertation. Leppälä [1995] provides for a conceptualisation of contractual R&D at VTT and evaluates its success and failure criteria, but does not investigate causes or effects of changes in competence.

Tiittula [1994] has carried out a longitudinal study of managerial changes at VTT, but addresses a very high organisational level of research institutes and VTT as a whole. It is difficult to make explicit individual projects, specific competence areas and their changes at that level – which would be needed to answer the second question. For making explicit and studying how competence had been created and how it had changed in inter-organisational projects during the past years, I have developed a framework, summarised at the end of Chapter 6. Miettinen [1996] has conducted a similar analysis of certain research projects at VTT, but more from a social than a business perspective.

Regarding the third question, it is obvious that projects per se do not affect competence evolution, but organisations and people involved in the projects. It is thus important to analyse why and how people act when conducting projects. This was one of the main reasons for revising the initial competence evolution framework I will present in Chapter 4. I adopted the concept of logic of action (cf. [Eriksson and Räsänen 1998]), somewhat related to the concept of managerial logic used by Tiittula [1994] when studying the management of VTT, but applied at a lower level of organisational groups. By using the concept, I have made explicit several effects of differing objectives, goals, and actions of groups of people on certain relationships, and thereby on competence evolution.

I have approached the fourth question by suggesting a relationship-based model for competence management and evaluating it against the literature and the two cases of my research. However, strategic management of competence for the business needs of a contract research organisation, using the portfolio of its projects, would deserve an additional study. I have only taken the first steps towards that direction.

I have utilised and extended some of the key concepts of the network approach to industrial relationships, and the resource-based view to firms. I have associated them together in the form of a competence evolution framework. However, the main emphasis of my work has been on the use of the framework. In other words, the core of my research involves the empirical investigation of competence evolution in certain projects from the mid-eighties to the present date. In this respect my research is related to Michelsen [1993], Tiittula [1994], Leppälä [1995], and Miettinen [1996], who have all been motivated by gaining a better understanding of what contract research is all about.

1.4 DISSERTATION STRUCTURE

This dissertation consists of four parts, not only forming a logical structure, but also illustrating the research process I have carried out:

- Part I, “Setting the scene”

The background of the study is outlined, the context of contractual R&D carried out in the form of projects by VTT is described, and the research problem is discussed. In addition, the research design I have followed is presented and related research is analysed. Additional pieces of related research are discussed in subsequent parts of the dissertation, to describe and evaluate the competence evolution framework.

- Part II, “Substance of competence”

The second part focuses on studying what competence means in the context of contractual R&D projects. The initial version of the competence evolution framework is presented and applied to analyse the fault diagnosis case. Preliminary empirical findings, initial theoretical contributions and the first managerial implications are also provided. How to associate the so-called inner and outer contexts, a key problem in contextual analysis, is discussed. The second part is based in part on [Seppänen et al. 1998a].

- Part III, “Competence evolution in focal nets”

The third part is at the core of this research, addressing how does competence change in R&D projects and what aspects of project relationships affect the changes. The competence evolution framework is revised and applied to analyse the code generation case. A cross-case analysis is also conducted, to compare empirical findings and evaluate the framework. The third part borrows from [Seppänen et al. 1999a].

- Part IV, “Strategic competence management”

The fourth part discusses experiences related to the findings of my research, seen from a managerial perspective. The focus is on strategic planning and management of competence within the context of external relationships. The results of the whole research are also summarised and discussed.

- Appendix 1, “Fault diagnosis story”

- Appendix 2, “Code generation stories”

Two appendices are included in the dissertation, to provide for an overview of the cases analysed. The appendices are included in their original form, as they were prepared by myself and annotated by some of the interviewees. The competence evolution framework is used to structure and analyse the case data in the second and third parts of the dissertation. More detailed data of the fault diagnosis case can be found in [Seppänen et al. 1998a] and of the code generation case in [Seppänen et al. 1999a], both available in electronic form at www.vtt.fi/inf.

2 RESEARCH DESIGN

This chapter describes the design of my research, especially addressing the process through which it proceeded. However, I will begin by shortly introducing my professional background. I consider this relevant for I am one of the participating actors in both of the cases studied.

I joined TKO as a researcher in 1983 and continued to work at ELE from 1994, mainly as a group and department manager. Because of this, I have an insider's managerial view of the phenomena I have studied. From 1994 on to Spring 2000 my work has involved managing Embedded software, one of the departments in VTT Electronics. Previously, research groups were called sections and they were organisational sub-units of VTT research laboratories. I managed the Software engineering section of TKO from 1987 to 1993, except in 1989 and 1992 when I was on leave.

The roots of this research include also a practical total quality management project I conducted as part of my executive MBA studies in 1993–1995 [Seppänen 1995]. As I had been an engineering researcher since 1983 and a middle manager since 1987, I thought of taking a pragmatic view of my continuing business studies as well. I carried out several Internet searches, trying to find out relevant questions and information on competence management in the context of contract R&D, as illustrated in Frame 1. However, the method proved to be unprofitable.

7 documents match your query.

1. Chapter 2, Section 3: Japanese Technology Commercialization Efforts
JAPANESE TECHNOLOGY COMMERCIALIZATION EFFORTS. The following four examples provide unique insights into how Japanese companies successfully bring products.
http://itri.loyola.edu/ep/c7s3.htm – size 20K – 20-May-97 – English
2. The Weir-Jones Group of Companies
Founded in 1971, the Weir-Jones Group of Companies has extensive capabilities in the design, testing, and analysis of marine and ground transportation...
http://www.weir-jones.com/ – size 4K – 19-Oct-96 – English
3. Chemistry Related Companies
Chemistry Related Companies. Last updated () 15-May-1997. CONTENTS: Analytical Services. Chemical Companies. Computers and Software. Instrument...*http://chpc06.ch.unito.it/chem_companies.html – size 38K – 15-May-97 – English*
4. No Title
ITI Industry Briefing focuses on public-private partnership. ITI completed its third successful annual Industry Briefing on 26 September 1995 ...
http://www.iti.gov.sg/iti_RnD/pub/innovator/issue20/indbrief.html – size 2K – 21-Dec-95
5. Pasu's Personal Information
Is this a picture of me? Pasu Decharin. Or this!! Date of Birth 15 November 1968. Nationality Thai. Marital Status Married with two chirdrens. Postal...
http://phoenix.acc.chula.ac.th/~fcompdc/pinfor.htm – size 5K – 7-Apr-97 – English
6. Keyword list
Keyword list. ABSENTEEISM ACCIDENT AND HEALTH INSURANCE ACCIDENTS ACCOMMODATION ACCOUNT MANAGEMENT ACCOUNTABILITY ACCOUNTANT ACCOUNTING ACCOUNTING AND...
http://www.nijenrode.nl/library/services/keyword.html – size 39K – 20-Jan-97 – English
7. publications – long
Publications by Kari Leppälä 1974–1996. SELECTED ABSTRACTS. A full paper is available on-line in postscript format, where indicated. VTT's own.
http://www.ele.vtt.fi/people/kari.leppala/kpl-abst.htm – size 29K – 2-Oct-96 – English

Frame 1. Internet search for “core competence + contract research”.

2.1 OVERVIEW OF THE RESEARCH PROCESS

In search of the research approach

I produced my first research proposal in Spring 1997 (an extract shown in Frame 2). The topic of the research was "maturity improvement model for contract R&D", involving "the assessment and improvement of maturity, by identifying (1) measurable elements of maturity, (2) maturity levels based on the values of these elements, and (3) procedures needed to review and reach a specific level". The model would be "implemented as part of a management information system". The topic, problem, and research design have since March 1997 changed considerably. I started from a managerial problem to be solved by constructing a prescriptive model and ended up with trying to understand competence changes in project relationships, analysing two cases in a longitudinal interpretative study.

Even after I had chosen the qualitative and longitudinal research approach, the method was not unproblematic since for a rather long time it had been my intention to investigate how competence evolution affects customer relationships. The titles of my first publications on this research ([Seppänen et al. 1998a,b], "Competence-based evolution of contractual R&D relationships") indicate this quite clearly. Later I focused on a reversed problem, the evolution of competence due to changing project relationships.

Maturity Improvement Model for Contract R&D:
The Case of Embedded Software

Research proposal, version 0.1-23, Date 3/31/1997
Veikko Seppänen, VTT Electronics

This is a proposal for doctoral research in economic sciences. The problem that I address involves the construction, validation, and deployment of a maturity model for research and development (R&D), based on the identification, utilisation, and continuous improvement of skills and technologies that, when mature enough, become the organisation's core competence implemented as core products and product platforms. The empirical part of this research focuses on contract research and development. Maturity is defined as the capability of the R&D process to produce effective and efficient results, in terms of the goals and criteria set by the R&D organisation itself and its shareholders.

The model includes both the assessment and improvement of maturity, by identifying (1) measurable elements of maturity, (2) maturity levels based on the values of these elements, and (3) procedures needed to review and reach a specific level. … The model aims at helping contract R&D organisations in identifying, maintaining, and extending their skills and technologies in the form of robust, core product platforms. … The model is implemented as part of a management information system. Among other means, the model is validated by an analysis of contractual embedded software R&D projects carried out by VTT Electronics from 1992 to 1998. Implementation of the model is empirically evaluated by a study involving a specific embedded software R&D case.

Key words: research and technology management, quality improvement, core competence, product platform, strategic planning, electronics industry, and software development

Frame 2. Extract from the abstract of my first research proposal.

Choosing the qualitative perspective

After a while I realised that my study aims at understanding a complex phenomenon by using a set of unified explanations, or a local theory. I chose to study three cases, i.e. fault diagnosis systems, code generation, and software process improvement, since they all involved project chains several years in length on certain R&D topics. This was a practical justification for choosing the cases, in addition to accessibility of documents describing the projects and availability and willingness of people who had taken part in them to serve as interviewees. From an empirical perspective, I considered the topics related to the three cases as interesting and illustrative examples of the R&D activities of the focal organisation.

A case was defined based on the projects dealing with a particular R&D topic. General managerial plans and reports of the focal organisation provided for additional contextual information. As a participating actor I had no difficulties in identifying the projects and the personnel involved. I started to collect case data by analysing documents and conducting the first interviews. The case data base kept increasing, and I encountered difficulty in finding an appropriate way of organising the data properly. I tried several approaches, such as classification based on the persons interviewed and the projects analysed, but I was not satisfied – the developments taking place were not to easy to follow.

Ghauri et al. [1995] compare qualitative and quantitative research, claiming that the main difference between the methods is the procedure, the techniques used to gather and analyse data. Qualitative methods, such as historical reviews, group discussions, and case studies, are flexible and unstructured. In a qualitative approach, research comes before theory and the main task of the researcher is to identify relevant factors and construct explanations, or the theory. I would like to argue, however, that the different procedures are a result from an even more important difference between the two research approaches, the nature of the data to be investigated. Ghauri et al. [1995] actually indicate this by stating that concepts, abstractions "representing an object, a property of an object or a certain phenomenon", are needed to build theories. A theory "may be viewed as a system for ordering concepts in a way that produces understanding or insight". I needed to identify and make use of concepts related to competence and projects, to understand, interpret and explain the phenomena involved in the three cases. A qualitative approach seemed thus to be well suited to attempt to solve my research problem.

Struggling with concepts

I spent quite some time with textbooks and writings on qualitative research, especially [Alasuutari 1994], probably one of the best known and most used textbooks on qualitative methods in Finland. According to Alasuutari, qualitative research is not the opposite of quantitative research based on surveys or statistical interviews. Quantitative analysis can be compared to classic natural science experiments, where hypotheses on how an independent factor affects a dependent factor are tested. Qualitative research aims at understanding and explaining real-life phenomena – by applying and/or producing theories.

Alasuutari summarises qualitative research having two main phases: *reducing observations* and *solving the problem*. The former consists of two different parts: analysis of the research material based on a certain theoretical and methodological viewpoint, and integration of observations alias qualitative analysis. My difficulties in organising the case data clearly involved the first part of qualitative research. I took a predominantly factual viewpoint to the case data, since I had an extensive collection of documents available. Although individuals interpreted these facts somewhat differently in interviews, I approached the key concepts of competence and relationships using mainly the information acquired from documents. According to Alasuutari it is appropriate to use typologies to analyse factual data. I constructed typologies based on the data gathered from the three cases, as is illustrated in Frame 3 and Figure 2. I even invented graphical notations to illustrate the concepts, as is shown in Figure 3.

Veikko Seppänen, June 1, 1997

Concepts in Core Competence R&D and Marketing Networks

The following concepts will be organised into an initial conceptual framework, and then evaluated and linked to a hypertext database of examples of concepts found by analysing the real case study research material, consisting of three different cases.

Modelling Concepts

structure:	function:	dynamics:
entity	context	life cycle
attribute	information flow (input/output)	phase
relationship	controlling flow (input/output)	state
	function	transition
	resource	event (condition/action)
	information store	time
	terminator (information source/sink)	

Core Competence Concepts

product	process
part	sub-process
application	process area
product requirement	process requirement
product function	process task
generic product technique	generic process technique
product technology	process technology

R&D and Marketing Networks Concepts

actor	activity link	communication
actor bond	norm	position
network	contract	co-ordination
role	knowledge	business exchange
activity	assignment process	need
organisation	outcome	time
goal	control	context
investment	commitment	cycle
resource	phase	resource tie

Frame 3. Ideas for classifying concepts related to the research.

SCOPE	personal projectual group-related organizational interorganizational	individuals project teams organizational units organizations networks
Basic level	**COMPETENCIES**	**RELATIONSHIPS**
ELEMENTS	applications functions generic techniques technologies	environment exchange goals and items resources (actors/others) activities outcomes (economic/social)
STATES	knowledge/skill core competence deliverable service core platform	evaluative (follower) contractual (project customer) co-operative (project partner) pivotal (key customer/partner) terminated (earlier collaborator)
CHARACTERISTICS	Content: tacit/explicit broad/narrow scarce/common dominating/supporting	Interactivity: capability mutuality particularity inconsistency
Operational level	**COMPETENCE MANAGEMENT**	**RELATIONSHIP MANAGEMENT**
PROCESSES	identification&acquisition (D) structuring&codification (A) deployment&protection (C) integration&maintenance (B)	establish enact expand terminate
LIFECYCLE	premarket competition exploitation networking extension of market share relinquishing	interest customership partnership commandship uninterest
EVENTS	competence required/rejected competence built/lost competence used/retired	activity started/stopped/finished resource acquired/allocated/lost exchange initiated/completed
Strategic level	**COMPETENCE STRATEGY**	**INTERACTION STRATEGY**
GOALS	generalization specialization combination extraction	competition dominance submission co-operation
CRITERIA	Contextuality: generic relationship-specific portfolio-specific platform-specific	Positioning: competitive dominated cooperative dominating neutral

Figure 2. An attempt to classify competence and relationship concepts.

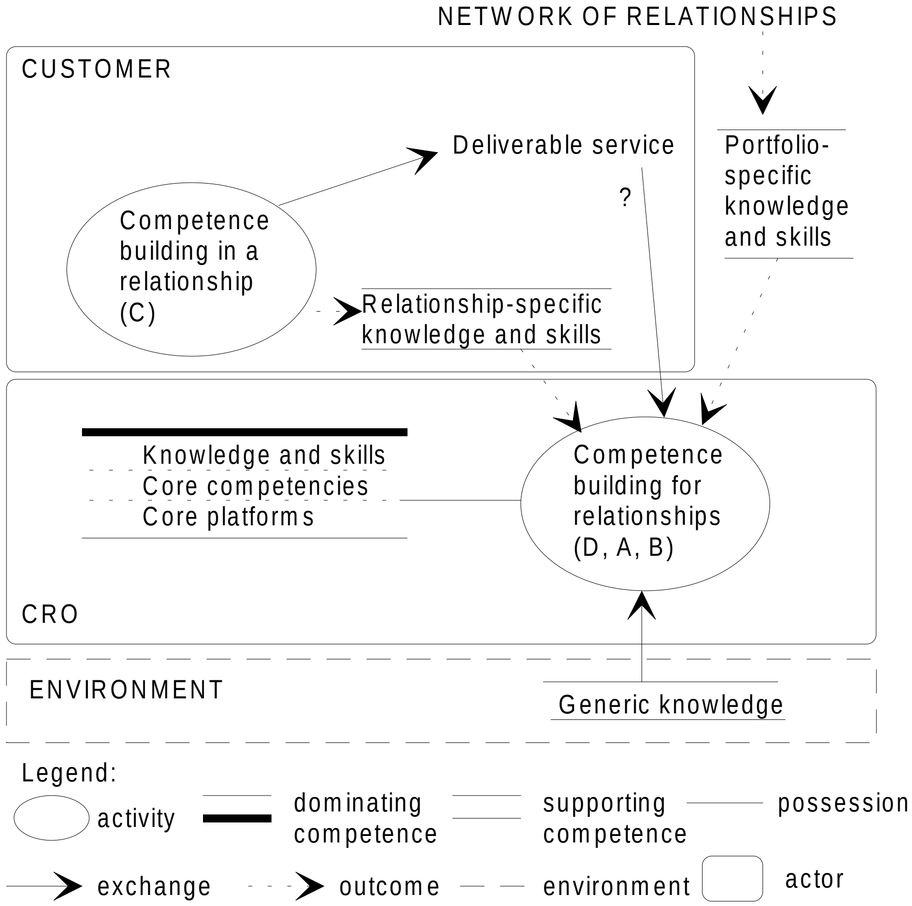

Figure 3. Notations for describing competence and relationships.

Without going into any details regarding the concepts included in Frame 3 and Figures 2 and 3, I wish to point out that the background of the concepts and notations was the technical topic of my research, software engineering. Software systems are often described by using concepts related to the overall context (Scope), the system itself (Basic and Operational levels), and the quality of the system (Strategic level). These kinds of concepts can be viewed from structural (Elements, Characteristics), functional (Processes) and behavioural alias dynamic (States, Life cycle, Events) perspectives. I therefore approached my case data much as a software engineering researcher, not as a business scientist.

After some time I, however, had to accept the fact that technically oriented concepts are far too constrained to describe complex phenomena carried out by human beings, be they groups or individuals. However, it is interesting to point out that [Seppänen et al. 1998a], published in Summer 1998, includes typologies of technical concepts, even a figure drawn by using software systems modelling notations. Moreover, some of the early process-related concepts shown in Figure 2 eventually became quite central in my research, although in a different form than I initially had in mind. Furthermore, the "core competence concepts" shown in Frame 3 later became very much a part of a typology of product-related competence elements. In any case, the "R&D and marketing networks concepts" shown in Frame 3 illustrate the difficulty of trying to apply generic concepts to a specific context, such as R&D, without any theoretical viewpoint tying the concepts together.

Gathering of both facts and feelings

After I had become somewhat frustrated trying to classify competence and relationship concepts, I again focused on collecting and analysing the case data. There are two kinds of material in qualitative research by which the concepts for study can be identified: the first existing independently of the researcher, and the second, acquired by the researcher him/herself. The latter may include material concerning the acquisition situation, such as a group rehearsal. However, it is important to recognise that the researcher has to interpret both the naturally occurring data and the purportedly acquired data. Critical evaluation of data sources is therefore needed to ensure that the data is usable and reliable enough for qualitative analysis and problem solving. Cross-referencing between various sources of information can be used to evaluate the validity of the data.

According to Alasuutari [1994] there are two basic approaches to ensure the validity of personally acquired data, such as results of interviews. The mechanistic approach tries to avoid informing the target of the study in order to obtain objective information. The humanistic approach tries, on the contrary, to establish a close and confidential relationship with the target of the study, to better understand various aspects related to the data and the roles of the people involved. Since data acquired in a confidential relationship is methodologically no more valuable than any collected in more indirect ways, I followed a mixture of the two approaches. I told my informants that I am conducting research on past VTT events. For the VTT managers and the focal researchers I explained that I was acquiring data to conduct doctoral research. This was important in the code generation case, since I knew that the two groups had conflicting viewpoints.

The second phase of qualitative research, problem solving, requires answers to “why” questions, and sometimes to “how” questions [Alasuutari 1994]. In cases where the target of the study includes abstract and conceptual matters, such as in mine, it is also necessary to ask “what” questions. One of the basic means of finding answers to these questions in qualitative research is to acquire and interpret individuals’ opinions on the subject matter. I acquired opinions on my cases by interviews, a reactive method of data collection. Since interviews aim at acquiring information, they are goal-oriented and pre-planned activities. An interview can be carried out in a very structured manner, based for example on a questionnaire. An unstructured approach leaves more room for variations in the conduct and content of the interview. Focused alias semi-structured interviews, which I was mostly using, is an intermediate approach. Hirsjärvi and Hurme [1995] point out that they already originate from the time of Aristotle.

One way to classify questions is a division into questions of fact and opinion. The former include both commonly known and private facts, the latter address the feelings, attitudes, and values of the interviewees. I did not plan for any fixed number of these two question types, but certainly the former were more common and the latter more revealing – as is clearly indicated e.g. in Appendix 2, where the feelings of the focal VTT managers and code generation researchers are contrasted to each other.

I still realised that the written data was the best source of factual information, not the interviews. Since my context was contractual R&D projects, plenty of well-organised documents were available for study. I tried to organise the "actual" concepts into three broad classes – R&D, management and strategy (Frame 4). This proved also to be rather unfruitful, even though it was useful for identifying the basic facts of the phenomena on which I was acquiring the data. For example, the project portfolio of VTT is described by the class of concepts of the type "project".

Veikko Seppänen, 31.7.1997

ACTUAL PROJECT RELATIONSHIP AND COMPETENCE CONCEPTS AT VTT/ELE

(...) means "is a" .../... means "and/or"

------------------------------------ "research and development"--------------------------------------
customer (company, consortium): manager, R&D person, support person
customer's customer (company: manager, consumer)
customer's sub-contractor (company, research institute: researcher, manager)
partner (customer, research institute)
funding body: representative, consultant
project (VTT's own strategic project, joint partnership project, and customer project)
project group: project manager, project member
project resource: quality system, information system, library, R&D facility, R&D tool
support person:
cost accountant, librarian, information system specialist, secretary, quality specialist
support resource: cost accounting system, other support information system
VTT's line management: research director, department/group/marketing/economic manager

R&D funding: VTT's budget, public financing, customer's financing
R&D policy (survey, pre-study, research, product development, consultation)
R&D source (VTT/customer/partner/public, existing product/technical solution)
R&D target (application, function/problem, engineering technique, and electronic technology)
R&D task: information gathering, research, reporting, and information dissemination

organisational R&D result: draft/proposed/accepted report, product/technical solution, patent
organisational financial result: expense, income, and net financial result
organisational associated result: customer/internal feedback, evaluation, project brochure
personal result: dissertation, academic degree
personal financial result: salary raise, bonus salary
---"management"--
project offer
project proposal
contract
project plan: schedule, budget, task (management, R&D), expected result, resources
project management group:
customer, partner, VTT's line manager, VTT's project manager (secretary)
project preparation/marketing task
project contracting/re-contracting task
project planning/re-planning task
project management task (control, cost accounting, quality review, and feedback acquisition)
project management group meeting
agenda/minutes of project management group meeting:
technical status, managerial status, decisions
managerial report (intermediate report, final report)
---"strategy"--
marketing plan (R&D department's, R&D group's): project marketing description/reference
annual plan (R&D unit's, R&D department's, R&D group's):
project planning description/reference
strategic plan: program reference/planning description, project reference/planning description

Frame 4. "Actual project relationship and competence concepts".

Choosing the theoretical viewpoint

After a while I simply chose a chronological classification for both the acquired and naturally occurring data as a common sense solution. The main justification for the choice was that my study was longitudinal. Finally, things started to converge. The chronologically organised data helped me to understand that I had to look for recurring patters over time. In order to find such patterns, in a rather brute-force manner, I combined the ARA (actor-resource-activity) network concepts [Håkansson and Snehota 1995], and a typology of R&D resources and activities based on [Rosenbröijer 1998].

Frame 5 shows an extract of an early plan to adopt the ARA model, with the key justifications for its use, referring first to a comprehensive analysis of science and technology studies by Reger and Schmoch [1996], who "refer to Håkansson and suggest that the relationship model should be used in modelling and evaluating R&D collaboration". It was my plan that "Abell's extended market model is used as the *a priori* competence model", so that this and the activity-resource-actor model would be "integrated by viewing the resource dimension of the ARA model as competence (or rigidity, in some cases) in its various forms". I also realised that the "other two dimensions of the network model [activities, actors] ensure that competence is addressed from the viewpoint of all actors and activities involved".

> Veikko Seppänen, 12.9.1997, draft
>
> **CORE COMPETENCE – NETWORK MODELING**
>
> Håkansson's ARA model is used as the *a priori* model for describing R&D relationships and networks. Relationships are formed in the ARA model of activity links, resource ties and actor bonds, networks are formed of activity patterns (which is done jointly), resource constellations (which are shared or exchanged), and webs of actors (who are acting together). A good justification for the choice is [Reger and Schmoch 1996]: … Reger and Schmoch refer to Håkansson and suggest that the relationship model should be used in modelling and evaluating R&D collaboration. Moreover, they discuss core competence and claim that it should be operationalised in the context of R&D collaboration. Abell's extended market model is used as the *a priori* competence model, so that the original three modelling dimensions of customer applications, customer functions, and product technologies are likened to application competence, functional problem solving competence, and technological competence. In addition, the fourth dimension of engineering disciplines is adopted.
>
> This is my original idea, but is partly supported by [Bogner and Thomas 1996]: … According to Bogner and Thomas, a company must integrate unique product attributes with more general, commonly available attributes. Markets can be viewed as Abell, so that competition is based on which needs the bundles of attributes respond to, and which technologies (or skills) provide them, in comparison to others on offer. Narrow focus (one dominating attribute), mixed skill (one dominating and several supporting attributes), and broad aggregation strategies are possible (evenly important attributes). Elfring and Baven address two of the dimensions from the capability building viewpoint, functions and applications. [Elfring and Baven 1996]: … According to Elfring and Baven, capabilities can be classified into functional and application capabilities. The former are organisation skills related to its function, the latter are needed to adapt the functional capabilities for the needs of specific customers. The two models are integrated by viewing the resource dimension of the ARA model as competence (or rigidity, in some cases) in its various forms. The other two dimensions of the network model ensure that competence is addressed from the viewpoint of all actors and activities involved. For core rigidities, see [Leonard-Barton 1992].

Frame 5. A plan to apply the ARA model in my research.

I found out that time and changes of competence and project relationships must somehow also be modelled. This notion brought me to writings on process-based analysis of industrial relationships, such as [Pettigrew 1997]. It also became apparent that there was very little material available on process-based analysis of competence. According to Pettigrew, a process is "a sequence of individual and collective events, actions and activities unfolding over time in context". The time quality of process analysis "lies in linking processes ... to the location and explanation of outcomes". Pettigrew claims that "writing a case history can help in getting on top of the data".

I wrote a history for the cases, by producing summaries of my initial understanding on what had happened. I tested my analytical vocabulary, as Pettigrew calls it, in a single case first. Frame 6 illustrates some of my initial thoughts on this vocabulary, which later resulted in the typology presented in [Seppänen et al. 1998a]. The fault diagnosis case that I addressed at first involved a rather compact set of projects and resources from the viewpoint of VTT, which made it not overly difficult to deal with.

Veikko Seppänen, 12.9.1997

Activity entities:
Activity relations:
Activity attributes and their values: role (dominating/supporting)
Attribute value definition:
Resource entities: knowledge, skills, capabilities, competencies, core platforms, core products, service specifications, service results, supporting resources, time, money
Resource relations:
Resource attributes and values: form (tacit/explicit), coverage (wide/narrow), availability (scarce/common), role (dominating/supporting), content (application/ function/ discipline/ technology)
Attribute value definition:
Actor entities: own actors, competitors, partners, customers, funding bodies, others
Actor relations:
Actor attributes and values: aggregation (individuals, project teams, organisational units, organisation), task (development/management/marketing/end use/other)
Attribute value definition:

Frame 6. Early ARA-based analytical vocabulary.

Conducting the research – a longitudinal case study

After I had written a working version of a report on the fault diagnosis case [Seppänen et al. 1998a], I realised that in order to create a well-grounded local theory based on findings from the case data; the egg, I would have needed to throw away the chicken; the first version of the competence evolution framework, produced mainly on the basis of related research. Since the framework had made sense when analysing the case data, I did not want to demolish it. Therefore, I spent the year 1998 by applying the framework to the code generation case, by using the case study method, as will be discussed in more detail in the next section. However, it soon became obvious that I needed to revise the framework. Moreover, I realised that it is not the sole amount of empirical case data that matters in qualitative research, but the understanding and interpretation of the data. I thus dropped the software process improvement case, it involving the shortest chains of projects, starting from the early-nineties, and focused my attention on the other two cases.

I managed to revise the framework, by using a kind of abductive reasoning process of interleaved theory building and experimentation, go through the code generation case, and publish the results [Seppänen et al. 1999a,b]. I had considered the idea of analysing the fault diagnosis case anew with the revised framework, but in that option the true evolution of my framework would have remained less visible. Instead, I decided to follow the approach used, for example in [Halinen 1994], to present both versions of the framework and describe how and why the initial framework was changed.

However, Halinen presents an initial model for the development of professional service relationships, based on literature and related research, and an improvement of the same model based on the analysis of a single case. I present an initial framework based on literature and related research, and use it to analyse the fault diagnosis case, where the focus is on the content of competence. Competence evolution in the context of contractual R&D project relationships is then analysed by the altered framework, using the code generation case. In this case, competence did not evolve as the different parties involved would have wished, as opposed to the fault diagnosis case. However, this enabled the carrying out of a hopefully insightful cross-case analysis. The analysis compares both the cases and the successive versions of the framework.

In this research networks of relationships are used to understand competence evolution. As I pointed out above, they themselves have been the topic of research. After having read several dissertations published in the nineties, it seems that researchers have not changed radically the core network concepts, but preferred to add new concepts on top of the network core. [Holmlund 1997] and [Tikkanen 1997] are good examples of such studies. Holmlund extended the ARA network model by quality concepts, Tikkanen by process concepts. There is nothing wrong with this kind of approach, it actually indicates the increasing maturity of the network paradigm in business studies. I also followed the same path by linking competence and network concepts together via the ARA model. One example of the extensions I made to the model, as will be explained in the second part of the dissertation, is the use of colours in relationships to be able to describe VTT's project portfolio. By using the network model as an analytical tool, I managed to make explicit and understand competence concepts and their changes in the context of my research, contractual software engineering related R&D services.

Axelsson [1995] claims that "one of the foremost issues in future work on this subject [industrial networks] will, in all probability, be a continued development of the basic concepts and content". Up till now, researchers have been focusing mostly on different ways of describing and characterising networks. According to Axelsson, "research urgently needs longitudinal descriptions of different industrial networks so as to categorise them, and also to understand the way they operate in terms of reaction patterns, the impact of specific actors, propensity to changes, etc." I started this research from the categorisation viewpoint, but found it rather limited and passive. In comparison, the process-based approach that I followed in the second case study revived the network model that I was applying.

Easton [1995] points out that “in practice, many if not most industrial network research studies have adopted, consciously or unconsciously, a case approach”, because it “allows the detailed study of particular events over time by different forms of data collection, so that the dynamics of industrial network processes can be teased out”. Halinen and Törnroos [1995] describe how this could be done, by discussing three different methodological approaches: historical or retrospective studies, follow-up studies, and futures studies.

I clearly made a retrospective study, like many other researchers. However, the revival of networks in my second case did not merely involve a longitudinal analysis of changes in networks. It also addressed the way that network activities could be understood as comprehensive business processes where people and organisations are involved. I owe this view to [Tikkanen 1997], which seemed thoroughly reasonable from the very first reading. Easton states that “the potential for action research in industrial networks is not high at this stage of our understanding”. However, I would claim that the potential of people who participate in everyday industrial networks, such as myself, has not yet been fully exploited for the purpose.

To summarise, the research approach I followed can be characterised as a qualitative and longitudinal study of competence changes, where competence dynamics is investigated in the context of evolving project relationships. For this reason, I carried out a retrospective analysis of R&D projects of the focal organisation involved in two different cases. The analysis was done using the case study method, based on gathering and interpretation of naturally occurring documentary data and purportedly acquired interview data, as will be discussed in the next section. Although the empirical analysis of competence changes is at the heart of my research, I also build the theoretical framework that was used in the analysis.

2.2 CASE STUDY METHOD

I got my hands on the textbook on case study method written by Yin [1988] in June 1998, and became convinced that I had found the right method for my research. Afterwards, I have learnt that what Yin proposes is a rather mechanistic approach to case study research, as he emphasises the work procedure at the expense of, for example, methods used to understand and interpret case data. However, [Yin 1988] suited well for my research, because I needed a frame especially for conducting the phase of the qualitative research process where observations are reduced [Alasuutari 1994]. Such textbooks as [Smith 1981] on business studies appeared as far too elementary in this regard.

For solving the problem, the other main phase of the qualitative research process, I was developing the competence evolution framework. This framework was fitted to the specific context of my research, as opposed to such general approaches to qualitative data analysis described, for example, by Bryman and Burgess [1994] and Silverman [1993].

According to Yin [1988], a case study strategy is relevant when the research question is of the explanatory form “how” or “why” type. In addition the research does not require control over the behavioural events being studied, and the focus is on contemporary events. I would like to add that it is important not only to track behavioural events, but also to understand and interpret them.

Yin defines a case study as “an empirical inquiry that investigates contemporary phenomenon within its real-life context; when the boundaries between the phenomenon and context are not clearly evident, and in which multiple sources of evidence are used.” I was interested in comprehending how competence emerges – or withers – and how it is being exploited within the context of relationships of a contract research organisation. The boundary between this phenomenon and its context is all but clear-cut. As an example, from a managerial viewpoint I was also interested in knowing how to control the context, e.g. how some competence should be managed to create new relationships. My focus was on understanding events that took place during the past several years. In other words, I addressed interpretation of contemporary events. As an investigator there was no need for me to have any control over the events, but neither was I an outsider – I was one of the participating actors in both of the cases that I studied.

Since the conditions for carrying out a case study suited my research, and participant observation is not excluded from the method, I followed the case study approach. A longitudinal perspective is necessary to understand and explain changes in real-life phenomena. In my two cases chosen from the three original alternatives as discussed above, I investigated a period of over ten years, from 1985 to 1998, including several fault diagnosis and all code generation related projects that had been carried out by VTT. From the viewpoint of the focal organisation under study, the contemporary events related to the cases were thus well covered.

A case study research design includes at least the following elements: research questions, propositions (if any), units of analysis, the logic of associating case data with the propositions, and the criteria used to interpret the results. In addition I needed to use some patterns to structure explanations. To solve the *research problem*, “How does competence evolve in R&D project relationships?”, I addressed the four specific questions listed in Section 1.3.

A research *proposition* is a means to focus the study on a relevant topic. The main proposition of this research is that if the skills and capabilities of a contract research organisation evolve towards core platforms [Hamel and Prahalad 1994], this enables the organisation to be efficient enough with regard to its capabilities and flexible in its relationships. A foundation for the proposition is that competence indeed does emerge, not only for customer needs within an organisation but as part of inter-organisational activities. A *unit of analysis* specifies the elements to be studied, usually on the basis of how the research question has been formulated. The main units of analysis in this research are the resources and activities of a contract research organisation. The main *patterns of explanation* are projects, whose actors and relationships I have conceptualised in the form of focal nets.

According to Yin, the quality of the research design of a case study can be evaluated by using four tests. The first test is *construct validity*, for which multiple sources of evidence, chains of evidence, and informant review of the case study report can be used. Secondly, *internal validity* by pattern matching, explanation building or time-series analysis can be ensured. The third one is *external validity* by replication in multiple case studies, and finally, *reliability* based on a documented case study protocol and case data base. One of the strengths of case studies is that evidence from physical artefacts and documents to interviews and direct observation can be utilised. I have used all these sources to pass the construct validity test.

Furthermore, the interviewed key informants reviewed draft case data summaries. Yin [1988] claims that "for case studies, the most important use of documents is to corroborate and augment evidence from other sources". In this research things happened the other way round. Within the context of a contract research organisation, managerial and technical documents are the basic means for professional people to communicate and agree over technical and managerial issues. Most research results are documented either as internal papers or as public reports, and many of them are also evaluated in written form. Problems in contractual projects are often noted in managerial documents. Individuals tend to keep personal diaries writing down, for example, other people's opinions on joint plans, activities and results. To try to structure this kind of data around interviews, which are "most commonly, ... of an open-ended nature" in case studies [Yin 1988], would be difficult, which I personally noticed. Therefore, I produced an initial case data base from a chronological set of organisational and project documents and corroborated this data with the results of interviews, to create a case data base.

I interviewed altogether some fifty people for the two cases. In the code generation case this included six out of the seven *focal researchers*, five *line managers of VTT* directly involved in code generation related activities, and seven *colleagues* associated indirectly with code generation projects. In addition, ten VTT and industrial people who were *end users* of the code generator, a *sub-contractor* used in its further development, the *two representatives of the funding body* Tekes involved in some projects, and three *industrial customers* who contracted code generation projects from VTT. Eleven members of the *management team of the* biggest code generation *project Sokrates* were also interviewed. The roles of these groups of people will be made explicit in Section 7.3 (cf. Tables 13 and 14) and analysed as part of the inner context of the code generation case.

The interviews were conducted both as face-to-face discussions, by phone, as a group rehearsal for the code generation researchers, and by using electronic mail. The group rehearsal was carried out in an unstructured manner. However, my own case summary had been sent to the informants beforehand. Most of the parties in the two cases were interviewed using a semi-structured approach, asking them to answer questions organised around the specific projects in which they were involved. A marketing student carried out a few interviews as a training assignment. As an example, Frame 7 illustrates some questions put to the members of the steering group for the Sokrates project.

Information was sought on the results expected and the results gained from the project, as well as about the main features of the project. All informants answered the questions, some with brief replies, others with quite detailed additional comments.

Questions for Sokrates steering group members, Spring 1998

1. What skills and results did you expect from the Sokrates project?

2. How was your contact with the project established?

3. What information and experience did you get from the project?

4. Did you utilise any of the results and if so, with what results?

    SA/SD-based design method
    Coding rules for the C programming language
    Code generator prototype
    Operating system nucleus
    Communication software package

5. What in Sokrates was irrelevant from your viewpoint?

6. Would you have needed code generation for other programming languages than C, or from other source languages than SA/SD design models?

7. Did you consider the use of code generation after the project, did you buy any commercial code generators, such as Prosac?

8. What do you think of code generation at the moment?

9. Do you still use the SA/SD method in design?

Frame 7. An example of semi-structured interview questions.

Hirsjärvi and Hurme [1995] estimate that the typical time spent on an interview is from one and a half to two hours, but that no strict guidelines have been laid down for their length. Some of my interviews only lasted from fifteen to thirty minutes, and most of the external parties just answered my electronic mail messages. Therefore, it seems to be the case that information technology is gradually changing interviews as an interactive process between the researcher and the interviewees, controlled in real time.

However, the problem of interpreting interview data cannot be overlooked. Interpreting electronic messages is possibly more difficult than interpreting interactively gathered observations. I did not use any interview summary cards or special records, since the problem of organising the case data was not that of arranging a folder of card files. It was rather a question of the chicken-and-egg problem. I was supposed to produce structured explanations based on the analysis of the case data, but the data needed to be organised into understandable structures to enable the analysis.

I wrote summaries of the two cases, based on my initial understanding of developments, and sent them to the key informants. The purpose was twofold; to help the writing of a more elaborated case history, and to make explicit my own managerial perspective on the cases as a participating actor, a manager and “key decision maker in an organisational setting” [Yin 1988]. The benefits of the summaries were considerable, especially in the latter sense and in the controversial code generation case. The informants could see on paper what one manager thought about the case and had the possibility to tell their own stories. Moreover, although I studied contractual project-based relationships and had thus no difficulties in recognising them from the case data, it was useful to learn how different parties considered the importance of certain projects. As an example, a financially very small project was mentioned by the focal code generation researchers as one of the best projects in the whole project portfolio.

My initial case summary helped to trigger extensive additions especially from the focal code generation researchers (cf. Appendix 2): two comprehensive, yet divergent written outlines of the same subject, an 18-page report of a group rehearsal that lasted for three hours, and a 90-minute interview with a colleague. The summary could be accompanied by the story of the code generation researchers, the “multiple actor-informants own constructions of their situation in the context studied” [Tikkanen 1998].

According to Yin [1988] “the most desirable option is to disclose the identities of both the case and the individuals”. However, if the study has been on a controversial topic, as is the case with the code generation R&D where the viewpoints of the key participants clearly conflicted, anonymity helps to protect the real case or at least its participants. Yin points out that “the anonymity of the individuals alone may be sufficient, thereby leaving the case itself to be identified accurately”. I followed this suggestion for both of the cases, although my initial summary of the fault diagnosis case (Appendix 1) did not result in considerably conflicting augmentation. Some minor controversies involved in this case are documented in [Seppänen et al. 1998a]. They mainly include certain customer opinions, and not as many internal disagreements within VTT as in the code generation case.

Several thousand pages of documentary data were retrieved on the two cases, mainly from the archive of the focal organisation. Figure 4 illustrates the different portions of the case data. In the front there is a row of folders full of related research publications, in the back a much larger set of documents, on projects involving the cases and the management of the focal organisation. The most extensive set of personal notes I studied included eight 200-page diaries from the mid-eighties to the late nineties, shown in the uppermost left corner of the figure. All information on the Finnish electronics and automation industries, related to the outer context of my research, is included in the bag to the left of the related research folders.

Other types of material were also available for investigation, e.g. the VTT code generator and its commercial counterpart, a small-scale model of an elevator for which software had been automatically generated, and a marketing video tape, to mention just a few.

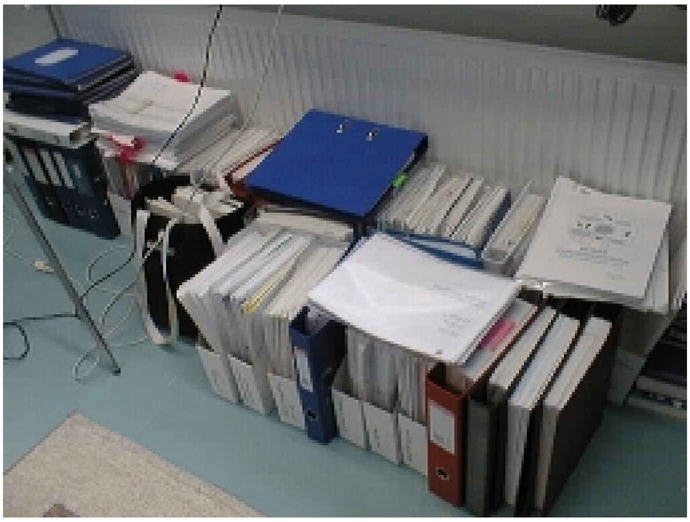

Figure 4. Illustration of the portions of the documentary data.

The data collection process for the two cases took altogether more than a year, resulting in a textual case database of about three hundred pages. This database includes extracts from the analysed documents classified in a temporal order, the case summaries with additions, the results of the code generation group rehearsal, as well as the free-formatted data gathered from the interviews. They "may be considered as the formal part of the database and not part of the final case study report" [Yin 1988].

The whole portfolio, 23 projects from the mid-eighties to the present date, of the code generation case was analysed, for fault diagnosis the five main projects were addressed. Analysis and interpretation of this case study evidence was done by utilising the competence evolution framework. The network part of the framework, i.e. conceptual models of contractual R&D project relationships, were used to build the explanations. I organised activities chronologically involving different actors and relationships over time, cf. [Dubois 1994], mostly in tabular form. Since the period of time I studied was less than twenty years and the archived project material included plenty of details, it was easy to extract the chronological relations of activities from the data.

Explanation building is always an iterative process in multiple case studies, as shown in Figure 5. I followed this kind of iteration cycle in my research, publishing two different case reports [Seppänen et al. 1998a, Seppänen et al. 1999a] as the main results. The basic iteration loop consisted of making the initial proposition based on relations to former theories, comparing the findings of the first case study to the proposition, revising the proposition, and repeating this cycle for the second case. However, I would like to stress that one of the most difficult aspects of the process was not to revise the proposition on how competence evolves in project-based relationships, but to build and revise the theoretical framework – patterns of explanations, as I have called it above. It was not built based on textbooks of qualitative and case study research, but on competence and relationship literature.

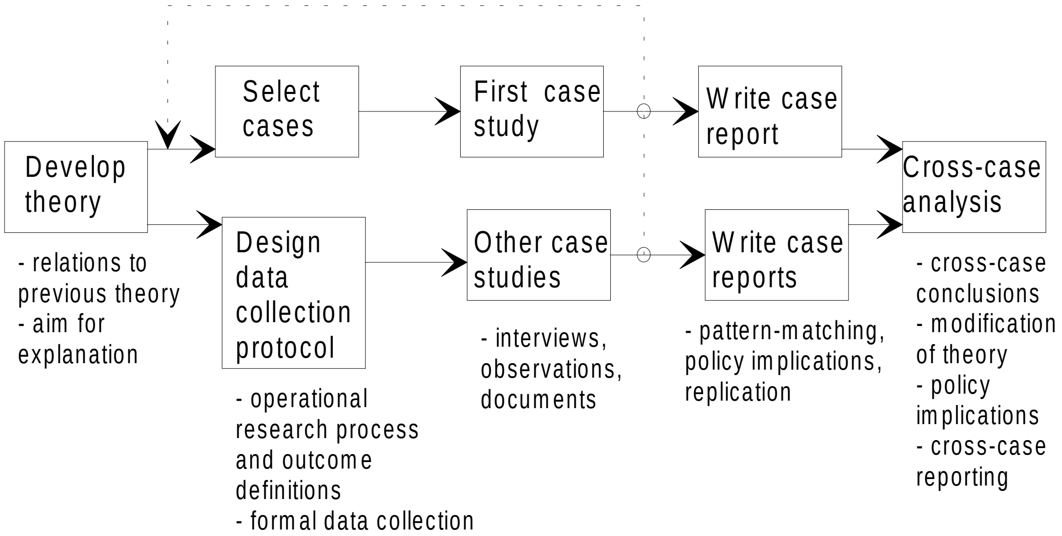

Figure 5. Case study method for multiple cases [Yin 1988].

According to Yin, it is appropriate to first conduct the analysis of the embedded units within each case, in order to consider the results as factors in pattern matching or explanation building at the level of the single case. Explanation patterns should be compared between cases, rather than between the results of embedded unit analysis. To explain the evolution of each respective competence, I followed this advice by analysing individual fault diagnosis and code generation projects, the most important embedded units. Explanation patterns in the two cases were preliminarily compared in [Seppänen et al. 1999a] and will be addressed again in this dissertation, to analytically generalise explanations of the research proposition.

My third case, software process improvement, was originally intended to help me to analyse how service products can be created from core platforms, since a specific product concept had been developed by the focal organisation for this purpose. However, since I did not carry out a full study of this case, I will not employ it even as an illustration in the fourth part of the dissertation, where strategic competence management is discussed.

My experiences from the case study method can be summarised as follows:

- identification of cases based on projects was quite easy,
- documents became a backbone of the case data,
- case summaries written early on were useful as stimuli,
- building of the theoretical framework to analyse the case data was the most difficult iterative process to manage,
- writing of intermediate case publications was very helpful.

The last item, the writing of altogether four scientific publications on the cases [Seppänen et al. 1998a,b, Seppänen et al. 1999a,b], was useful in two senses: the theoretical framework was exposed to initial peer review, and considerable portions of the case data could be published as parts of the reports already during the ongoing research process.

3 ANALYSIS OF RELATED WORK

One of the difficulties in analysing related work was that competence studies and network analyses have been conducted to a large extent in isolation from each other. The few exceptions include, e.g. [Easton and Araujo 1996, Alajoutsijärvi and Tikkanen 1999a, Rosenbröijer 1998]. They have addressed activities used to create and make use of resources in relationships, which is my perspective to competence in this research.

I will focus on competence studies in this chapter. Networks of industrial relationships will be discussed in more details in the second and third parts of the dissertation, where successive versions of the competence evolution framework are presented, based for the most part on existing research.

In the following, I will first make an overview on competence research from the viewpoint of this study, starting from the research carried out to conceptualise and analyse firm resources. I will then discuss how core competence has been characterised by different authors. Since I am interested in the evolution of capabilities from implicit skills to explicit competence, and am addressing a technical domain, I will devote one section to core products, platforms and product families. In closing, I will discuss studies of networks of industrial relationships to pave a road for the definition of the competence evolution framework in the second (Chapter 4) and third (Chapter 6) parts of the dissertation.

3.1 COMPETENCE STUDIES

Since the publication of Prahalad's and Hamel's Harvard Business Review article “The core competence of the corporation” [1990], the concept of competence has attracted a great deal of attention. From the viewpoint of business studies, the term is related to the so called resource-based view of companies that had been introduced earlier (cf. [Pfeffer and Salancik 1978]). Within the context of R&D, it has involved e.g. industrial development of new products. A good example of this stream of research is [Lindman 1997] that was discussed briefly in Chapter 1.

Despite the fact that the roots of competence thinking are in the well-known resource-based view to firms, and that for example the textbook of Hamel and Prahalad [1994] has been extremely popular, it appeared neither to be straightforward enough to find common elements in competence literature nor existing meta-theoretical studies on these elements. I was not only looking for definitions of concepts such as assets, skills, resources, capabilities and competence, but also interested in how their changes over time had been made explicit, i.e. on competence dynamics. Sanchez et al. [1996a], titled “Dynamics of competence-based competition”, became a key piece of related work. Although the word “dynamics” refers to competition and not to competence, the textbook includes several articles addressing competence changes. Moreover, a meta-theoretical analysis of competence by Lewis and Gregory [1996] is included, and an evaluation of competence from a network perspective by Easton and Araujo [1996].

Lewis and Gregory [1996] make a difference between four views to competence: economics (cf. [Teece et al. 1992]), corporate strategic management (cf. [Prahalad and Hamel 1990]), management of technology and learning (cf. [Leonard-Barton 1992]), and human resource management (cf. [Sparrow 1992]). Based on the concepts of competence, core competence, distinctive competence and dynamic capability used by these authors, Lewis and Gregory provide their own working definitions for the first three concepts, as is shown in Table 1.

Table 1. Views to competence [Lewis and Gregory 1996].

1. Economics: Teece et al. 1992
A. Competence: When firm-specific assets are assembled into integrated clusters spanning individuals & groups enabling distinctive activities to be performed.
B. Core competence: Those critical to a firm's survival. Should be derived with due reference to opportunities & threats facing the firm.
C. Distinctive competence: The differentiated skills, assets and organisational routines allowing a firm to co-ordinate activities that provide the basis for competitive advantage in particular market (s).
D. Dynamic capability: The capacity of a firm to renew, augment and adapt its core competence over time. Capabilities thus reflect the firm's latent competences.
2. Corporate strategic management: Prahalad & Hamel 1990
A. Competencies: The collective learning in the organisation, especially how to co-ordinate diverse production skills & integrate multiple streams of technology.
B. Core competencies: Source of competitive advantage. 3 qualifying tests: access to wide variety of markets; contributes towards perceived customer benefits; is difficult for competitors to imitate.
3. Management of technology (learning): Leonard-Barton 1992
B. Core capability: The knowledge set that distinguishes and provides competitive advantage. Identifies 4 dimensions.
4. Human resource management: Sparrow 1992
A. Competencies: Should be viewed as behavioural repertoires which some people carry out better than others.
B. Core competencies: Those that remain important to a firm. It is these competencies that provide continuity.
D. Transitional competencies: Those not currently important, not implied by the strategic plan, but change may only be implemented through their greater emphasis.
WORKING DEFINITIONS
A. Competence: Those activities which a company recognises as containing unique resources.
B. Core competences: Those competences that management perceive as of central importance to the company's goals and strategy.
C. Distinctive competences: Those competences that are recognised by the market and hence provide the basis for the organisation's competitive advantage in a market or markets.
D. --

As discussed below, I have adopted a view combining some aspects of dynamic capability included in the economic viewpoint, as well as organisational learning related to the technology management view. However, I consider competence as an ability to create and make use of resources in contractual R&D, i.e. in the context of a special task. Moreover, as Kirjavainen [1997] points out (p. 11), “interest in organisational learning seems to have arisen in many discourses of business research simultaneously”. It is thus useful to make the discourse explicit. In this research the discourse is the network-based approach to industrial relationships, especially as described in [Håkansson and Snehota 1995]. In this discourse, competence can be associated with activities performed by network actors to create and use resources.

Although I spent quite some time with Hamel and Prahalad [1994] in the beginning of the research, it became obvious that the corporate strategic management view to competence did not fit well with the context of my research. The strategic competence management approach that I will discuss in the fourth part of this dissertation is not any corporate-level endeavour, but geared towards management of small organisational groups and portfolios of related projects. Similarly, although I discuss knowledge creation by individuals and organisations based on [Nonaka and Takeuchi 1995] in Chapter 6, my focus is not on human resource management and learning as a cognitive process (cf. [Kirjavainen 1997]).

3.1.1 Resource-based views to competence

The so-called resource-based perspective on companies has resulted in a number of studies on skills, capabilities, and competence. Foss and Knudsen [1996] have edited the book “Towards a competence theory of the firm”, which includes a historical review of the competence perspective by Knudsen [1996]. According to the review, the three themes central to the competence-based perspective are viewing of a firm from the perspective of endogenous growth, sustaining of existing advantages within an industry, and homogenous understanding of strategy. These themes emphasise the corporate strategic management view to competence that I have not applied in this research. For the same reason, [Foss 1997] was not much used in my research, as “a reader in the resource-based perspective”.

Although Christensen [1996] goes a bit further when analysing the technology base of the firm from “a multi-dimensional resource and competence perspective”, his taxonomy of scientific research assets, process-innovative assets, product-innovative assets and aesthetic design assets does not fit well with the context of my research either. The main reason, which can be traced back to the landmark of the corporate strategic management view to competence by Prahalad and Hamel [1990], is that company resources are defined at an abstract level and in a static manner. Instead, I have taken the viewpoint of Sanchez et al. [1996b], who describe capabilities as repeated patterns of action involving certain assets used to create, produce, and deliver products to the market. In this research I consider assets to be the resources created and used in relationships.

Based on this view, competence in contractual R&D can be defined as *activities taken to create and make use of resources by individuals and organisations in project-based relationships.* Håkansson and Snehota [1995], which I have used as one of the main sources for building the network part of the competence evolution framework, also point out that company's assets, especially its resources, are too often considered to be given entities. However, on closer inspection it is the use of an entity that determines if they are resources or not. A resource can therefore be viewed as the relation between the provision and usage of an entity, rather than just the entity itself. Resources are both results of and conditions for some tasks, such as contractual R&D. In other words, they are variable and not static. Resources can be changed in two fundamental ways, either by modifying their features (resource transforming tasks) or by changing the way or purpose for which they are used (resource transfer tasks).

Analysis and development of competence

The competence analysis process proposed by Lewis and Gregory [1996] consists of activity and resource analysis and a strategic process review. Activities are suggested for evaluation by using the following metrics: importance, performance, imitability, transparency, transferability, and replicability. Their scarcity, imitability, durability, retention, codification, embodiment, and importance all measure the uniqueness of the resources. Competence is identified as a certain combination of activities and resources and illustrated by using competence maps. However, the analysis process, as it is described by the authors, lacks conceptualisation of the context in which the analysed competence would be exploited. In comparison, in my research certain R&D tasks, projects and organisational groups form a specific context for competence analysis.

Easton and Araujo [1996] propose several groups of resource dimensions for study. Existence leads to resource creation, maintenance, and replacement activities. Evaluation is necessary for resource management. Relationships with actors concern controllability, accessibility, and tradebility. Activities ensure integrity, versatility, complementarity, and understandability. The authors point out that "the general product and process technologies used in the industry are widely known and available. They are described in technical journals in articles authored by leading technologists employed by competing companies as a form of promotion". Contextuality of resources therefore decreases when the resources become more general. Also I have adopted this view in my research.

Sanchez et al. [1996a] claim that when competence is built, qualitative changes of capabilities dominate, and when it is leveraged and maintained, quantitative changes dominate. Sanchez and Thomas [1996] characterise competence building as "the process by which a firm creates its strategic options" and competence leveraging as the process by which the firm "exercises its existing strategic options". They state that managing of competence building and leverage is fundamentally different. On the basis of this research, I disagree. Instead of being diverse, the two processes are highly concurrent, qualitative and quantitative changes are intertwined, and management of competence changes should be based on a uniform strategy.

As Bogner and Thomas [1996] correctly point out competence is a market-driven concept. In addition to the internal development of knowledge and skills, an external competitive process exists. Bogner and Thomas view attributes as traits of companies perceived by their customers. A company must integrate unique product attributes with more general, commonly available attributes. Markets can be viewed so that competition is based on the needs the bundles of company attributes respond to, and with which technologies (or skills) they are provided, in comparison to other offerings. Both a narrow focus based on one dominating attribute, mixed skills based on one dominating and several supporting attributes and broad aggregation strategies with evenly important attributes are possible.

Bogner and Thomas propose “plus only”, “central” and “threshold” types of attributes, each of which may be insufficient, sufficient, or distinctive from the customer’s viewpoint. Distinct multiple attributes may amplify each other, together forming the core competence of the company. Changes in attributes and attribute bundles may either enhance or destroy competence. This may involve a change in the importance of an attribute, and the level of its offering to a particular customer. I have approached the importance of competence from a somewhat different angle, by investigating how competence evolves and is viewed by different parties over time.

Elfring and Baven [1996] propose four stages for the evolution of knowledge-intensive services, such as engineering design and software development. Learning and capability development in these stages is characterised by external exposure (alliances, demanding clients), learning from clients, leverage of client relationships and development of internal organisational capabilities. Capabilities are classified into functional and application capabilities. The former are the skills of the organisation related to its function, for example software engineering skills in a software house. The latter are needed to adapt the functional capabilities for the needs of specific customers. For example, a software house might need knowledge of discrete parts production, if it serves a car manufacturer.

At stage one, functional capability is used in-house to serve a specific application. At stage two, the application can be leveraged to several customers within an industry, and upgraded by learning from the different applications. Application expertise generated from completed projects will be pooled. At stage three, the expertise is not limited to the context of a single industry. At stage four, cross-marketing of combinations of functional capabilities to different industries and applications can be done.

The evolution of the core fault diagnosis platform to be discussed in the second part of this dissertation largely resembles a process of this kind. Moreover, the platform is based on a functional view, which is one of the key aspects of the stages proposed by Elfring and Baven. Based on the results of my research, this gives a much more useful view of competence analysis than, for example, the abstract industry-specific competence, strategy-specific competence, and firm-specific competence classes proposed by Tallman and Atchison [1996]. These classes are used by the authors to characterise the evolution of “strategic configurations”, but only in a broad context, to identify industrial phases from innovation, entrepreneurship, growth, and shakeout to maturity.

Study of inter-organisational competence in electronics industry

Gressetvold [1998] is discussed here although it is a piece of ongoing research, since he investigates competence through the kinds of roles that different organisations have in the electronics industry to develop and deploy various innovative technologies including both tangible and intangible assets. As only a few organisations can by themselves possess all the necessary technologies in the electronics industry, networking with others and resource development within networks are usual. In other words, Gressetvold takes the inter-organisational rather than intra-organisational view to competence building and leverage.

Gressetvold characterises this himself by stating that "New products are becoming increasingly complex with combinations of different technologies, resulting in a need for different areas of knowledge. These general trends do strongly indicate an increasing need for co-operation in relation to resource development – and especially in the business areas characterised by rapid markets, with correspondingly high frequency of resource development" [Gressetvold 1998]. I do agree, since this was one of the main motivation factors for me to carry out this research. However, Gressetvold considers capabilities being the capacity to deploy, i.e. to use, resources. In my opinion, this perspective is too limited. Deployable resources must continuously be created. Moreover, critical product-related capabilities need to be associated with such supporting organisational and process-related capabilities as project management skills.

Gressetvold classifies deployable resources into tangible (generic, specialised, or co-specialised), know-how (codified or tacit), and legitimacy (charter-based or firm-based). The resource taxonomy I have created does not include legitimacy. Instead, I use the project portfolio of one company-based (green) and two charter-based (blue, red) projects. Gressetvold classifies competence-based resources in the VLSI domain further as EDA (Electronic Design Automation), application, circuit design, and component manufacturing-related competence. EDA-related competence is associated with the other three types of competence. One of the basic problems with this classification, in the context of my research, is that it mixes product-related competence with process-related competence. Moreover, Gressetvold sees design competence as company-specific, a view that cannot be assumed in my research, for many collaborative partners may very well have quite similar design skills.

Gressetvold mentions the role of lead users in the development of EDA tool competence, because such globally significant tool user organisations as telecommunication product manufacturers may affect the features of tools long before the tools are commercially available. This remark is worth remembering later on when the code generation case will be discussed. The opportunity of the present active developers and future lead users of code generators were apparently not utilised by VTT in the turn of the nineties, because no strategy focusing on telecommunication applications existed.

Longitudinal views to inter-organisational competence

Anderson’s [1994] dissertation addresses the evolution of a mechanical engineering company’s competence over 60 years, from the twenties to the nineties. The company has developed 120 different products. Anderson combines two theoretical perspectives, interaction and resource transformation. The former deals with the company’s business relationships and forms the context of the evolution. The latter involves competence based on the products developed and delivered by the company. The setting is closely related to my research. However, Anderson's goal was to make explicit the dynamics of the company’s transformation, seen through the change of its products and affected simultaneously by external and internal forces and their combined influence. Therefore, although the research was carried out as a re-constructive case study similar to my research, its purpose was somewhat different and the period of time that Anderson examined was considerably, four times, longer than in my study.

When investigating the reshaping of the focal company over sixty years, Anderson addresses relationships that are important for the development of the focal company’s resources, especially the ones involving buyers and suppliers, and the ones breaking off existing relationships. A company’s resources represent its technology, organisation, knowledge and relations of personnel, and financial strength. In all, Anderson describes resources as core competence, knowledge accumulated by the organisation.

This view explicitly emphasises codified assets the focal company can produce and make use of through sales to customers. This is understandable because of the very long period of time that Anderson addresses. For instance, information on intangible resources created and used in the thirties or forties may prove to be very difficult indeed to recover in practice. Apparently for this reason, Anderson focuses on the role of products for understanding the evolution of a company’s relationships. Products are in her view “the seller's ability to solve the buyer's problems”.

However, changes that can be identified by studying the change of individual products are rather specific. Anderson suggests that products “of major importance for the company” are core products representing core competence. She analyses networks of relationships built around products, “product networks”, to identify core products and competence. I have investigated project relationships for the very same reason. Even though these relationships involve rather abstract engineering R&D services, in many cases explicit software products were produced. Therefore, networks of project relationships could be called “service product networks”.

Anderson mentions the following core products of her case company: vehicles, electric motors and welding equipment, mining machines, deck cranes, and hydraulic motors. Every core product made it possible for the company to enter markets and industries where the company had little or no prior knowledge. The growth of the company was thus closely related to the core products. I also consider core competence to be a means of providing an expanding market share to its owner.

However, competence may very well either remain in a certain application area or be utilised in new areas, depending on the change of industries and the economy as a whole. For example, process automation applications that were the original target of the fault diagnosis competence, are still a valid domain for VTT despite entry into new application domains in the nineties.

Anderson claims that "core competence becomes apparent relatively early and is retained through the period". My research indicates the same, because functional skills that help building core competence are visible early on. Another observation by Anderson similar to my research deals with the transformation of resources. According to her, this mainly took place as a successive expansion of resources in relationships; "the process chiefly responsible for the industrial transformation is the company's competence conversion. The resource conversion does not occur in a vacuum but within the structure that the relationships together form and provide".

Rosenbröijer [1998] takes an approach to capability development that is based on usage and mixture of resources. He considers resources to be the basic elements of company capability, and points out that they are heterogeneous and dynamic, i.e. change over time. The levels of company (internal), relationship (external), and network (external) are used by Rosenbröijer to analyse the development and integration of the focal company's capabilities, when interacting with other actors.

He defines competence as "a combination of knowledge and experience in combination with a special task", whereas capability is the "ability and willingness to organise a mix of resources for productive activities, where resources are the basic elements of the activities in question". In my research, competence involves creation and usage of both tangible and intangible resources – although the latter dominate. However, the view of Rosenbröijer that competence can be considered as related to certain task is very relevant in my research, too. I have investigated resource creation and utilisation activities, i.e. competence, in the context of a special task: contractual research and development.

"Directed capability" is a concept used by Rosenbröijer to study resource interaction at a relationship level. He uses phases to describe the development of the focal company's resources at this level, but it remains unclear if the phases were defined for the needs of the research only. He also uses the concept of "connection function", but does not provide exact definition of its meaning. Rosenbröijer shows the development and mixing of resources in the relationships that he studies as a linear list, classified according to a resource typology. The actual contents of the resources are, however, not addressed in detail but described in such general terms as "wide product range", "sharing experiences", and "new warehouse".

He identifies seventeen critical investments of the focal company's resource mix over fourteen years, i.e. the grain size of his analysis is only about one resource item per annum. However, when analysing individual relationships he takes a closer look and lists more resources. Rosenbröijer uses a typology based on financial, physical, organisational, human, technological, and reputation resources.

I have applied and extended this typology in my research, keeping in mind the key differences between his and my context. For example, resources from all the categories are not equally important and do not become evenly mixed and integrated with other companies' resources in the R&D projects I have studied. Furthermore, the focal organisation I have investigated may make the same resources available in the same way to several partners in joint projects, as opposed to Rosenbröijer's context, where this is rare due to the unique division of labour between companies belonging to a network.

3.1.2 Characteristics of core competence

Although I have not taken the corporate strategic management view to competence, I consider the common-sense view taken to *core competence* by Hamel and Prahalad [1994] as relevant. According to them core competence has to be something competitively unique, extendable, and capable of creating customer-perceived value. Hamel and Prahalad simply characterise "core" as a synonym for "critical", when discussing competence. This view makes sense in the context of a contract research organisation offering R&D services in the form of projects. In particular, the characterising words of core competence can be interpreted as follows:

- "competitively unique": since the customers are usually technical experts themselves, the supplier seeks to offer customers something new,

- "extendable": since the supplier cannot serve only the specific needs of one or very few companies, its competence must be extendable for many customers, and

- "capable of creating customer-perceived value": as a contract research organisation, the supplier must provide value for its customers, for which they are willing to pay.

Uniqueness vs. codification

An important aspect of core competence related to the first item, uniqueness, is that distinction can be made between context-specific, portfolio-specific, and generic competence [Alajoutsijärvi and Tikkanen 1999b]. By means of its utility and scarcity, core competence must include something that is experienced as being simultaneously useful and difficult to acquire from alternative sources [Boisot et al. 1995].

The company's strategic architecture should indicate what new benefits or functionality will be offered to customers during the coming years, to avoid core rigidities. A strategic architecture is "a high-level blueprint for the deployment of new functionality, the acquisition of new competence or the migration of existing competence, and the reconfiguring of the interface with customers" [Hamel and Prahalad 1994]. An important issue in this regard is the level of codification of competence.

Competence, be it knowledge, skills or capabilities, may be tacit or codified [Boisot et al. 1995, Sanchez 1997, Nonaka and Takeuchi 1995]. Tacit knowledge is not articulated. It can usually be transmitted (the term “diffuse” is often used) only slowly in face-to-face situations and often to a limited audience, in conditions of trust and shared experience.

In contrast, codified knowledge can be transmitted much more rapidly and impersonally to larger audiences. By definition, generic competence is well articulated, while relationship-specific competence is typically tacit. Much of the current discussion on competence advances the proposition that the tacit knowledge of a company is more likely to be the source of a competitive advantage than articulated knowledge.

The argumentation for this notion is the following: if some knowledge is articulated, it can be understood by anyone, and as a result can be relayed beyond the individuals and their organisations, which in turn leads to the loss of any competitive advantage. I do not take this for granted, not because of barriers of diffusion (cf. [Sanchez 1997]), but because I suggest that in order to become core competence, skills and knowledge must be institutionalised. If they are not, their leverage in inter-organisational relationships may be hampered and ultimately wither.

Extendibility vs. institutionalisation

Regarding the second item related to core competence listed above, extendibility, individual, team-based, organisational, and inter-organisational competence can be identified, as has been pointed out by Eriksson and Ropo [1995]. In addition to individual capabilities, there are collective knowledge and skills built and used in collaboration. In this research, I regard these as elements of competence, rather than individual knowledge and skills. Depending on the context, team-based competencies may involve informal interest groups, project teams, and operational groups in the line organisation. In this research I have focused on the last two groups of individuals.

Companies can also be seen possessing certain competence as organisations – this is one of the basic ideas behind the corporate strategic management view to competence [Prahalad and Hamel 1990]. Furthermore, individuals and organisations involved in a relationship within an inter-organisational network can be interpreted as forming quasi-organisations with their own capabilities [Alajoutsijärvi and Tikkanen 1999a].

Mainstream management and marketing management literature emphasises generic “good management practices”, often presented as rather universal abstract capabilities that can be transferred across companies and industries. Theoretical studies, such as [Cooper and Kleinschmidt 1996] and practical surveys [SRI 1996] have been performed to identify good research and development practices, along the lines of general management literature. Leppälä [1995] investigates them in the project relationships of VTT. Tiittula [1994] addresses a broader managerial change at VTT as a contract research organisation that he characterises as "farewell to bureaucracy".

Customer perceived value vs. the quality of competence

The value of competence, seen from the customer's viewpoint, can in the context of contractual R&D be likened to quality. Remembering the proposition of this research of competence evolving in inter-organisational relationships, it is useful to view quality from the relationship perspective. For this reason I will briefly discuss the research of Holmlund [1997], as she has conducted a study on relationship quality.

Holmlund claims that "firms' quality perceptions are complex, context-bound and dynamic, and not easily transferred to other relationships and points in time". She suggests that service management, relationship and network perspectives provide for an elaborated enough viewpoint to quality. She uses the concept "relationship quality", associated not only with actor bonds but also with activity links and resource ties, by which I have conceptualised competence in this research.

Holmlund concludes that actor bonds are related to the social dimension of quality, whereas activity links and resource ties concern the technical dimension. The latter includes various process types and characteristics, such as reliability, innovativeness, use of competence, speed, use of physical resources, and flexibility and security, in addition to technical outcome. When she makes explicit the technical dimension of the quality, she presents instances of the process and outcome aspects, but fails to present how the two are interrelated.

In other words, she does not present the interrelationships of process related and outcome aspects. They are mixed together so that the interaction process, the exchange items produced and the characteristics of both the process and the exchange items are intertwined. Individual items can be taken apart and related together via the positive, negative and neutral views to relationship parties, but the interaction process is not made explicit. As a result, one cannot actually view the evolution of a relationship and see how different quality aspects were born and developing. In order to avoid this, I have analysed expected, perceived and historical values of competence over time, as they were seen by the different parties of the evolving relationships.

3.1.3 Core products, platforms and product families

Core competence has also been addressed from a rather technical perspective, on the basis of product families and platforms. I will discuss work on product families, since I am also addressing a technical domain in this research. A product platform is based on the idea of a family of several, similar derivative products that share parts of the same design. Meyer and Lehnerd [1997] define a product family as "a set of individual products that share common technology and address a related set of market applications", and a product platform as "a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced."

Companies focusing on designing new products one product at a time may fail to identify commonality, modularity, and compatibility in their offerings. Meyer and Lehnerd discuss Black&Decker as an example of the successful use of product platforms. A Hewlett-Packard ink jet printer family is discussed as a good example of product families. The platform architecture of the printer product family consists of mechanical, electronics, and firmware subsystem architectures and interfaces. The combination of subsystems and interfaces defines the architecture of any single product. Some interfaces may be strategic, be they industry standards or defined in-house. This is quite similar to Anderson's [1994] product-driven view of core competence, especially as the architecture is based on technologies and not on functionality. For example, the electronics subsystem of printer products includes “applications software”, which is not addressed in any functional details. The core part of the subsystem architecture includes “ink technology”, whereas in my view it should preferably be the functionality of "ink-based printing onto paper".

For software product platforms, Meyer and Lehnerd use the concept of “layers of computing technology”. The concept consists of applications (general-purpose and vertical), software development tools, operating systems and computer hardware. The generalised architecture of a software product platform includes add-in modules for creating derivative products, the “engine” and operating system services. Single, common interfaces are a key means to implement the architecture. Although the layers of computing technology divide the basic functionality of software products into four areas, they remain as abstract concepts. I have investigated the content of these kinds of platform layers in my research, using the four dimensions of applications, functions, techniques and technologies.

According to Meyer and Lehnerd, the development process of a product family, “a thought architecture” for effective product design, includes the following principles: product family planning and robust product platforms, simultaneous design for production, global product design and market development, and the discovery of latent and unperceived customer needs. The last principle is one of the key ideas in the core competence concept of Hamel and Prahalad [1994]. The so-called product family map shows the evolution of the platform and derivative products over time. The generic tasks involved in the evolution are development of the original product platform, platform extension, development of derivative products and development of new platforms, to create a “power tower”.

The power tower consists of common building blocks for product platforms (discovered and integrated), successive generations of product platforms, and market applications (market tiers vs. market segments) for derivative products. Building blocks of product platforms include customer insights that may be latent, product technologies for “the implementation of knowledge with the potential benefits incorporated into a product or service”, manufacturing technologies, and organisational capabilities that include distribution, customer support, and information systems.

This process resembles the evolution of core platforms in my research. However, Meyer and Lehnerd consider the process as an internal one affected by general market forces. I am more interested in the kind of ripple effects of forces and individual project relationships. In practice, the effects are caused by many distinct relationships between specific actors operating on the market and inside the company.

According to Meyer and Lehnerd, a platform strategy is “a top-down planning strategy to maximise market leverage from common technology”. In one of the cases studied in this research, market leverage could indeed be maximised by a core platform, but the strategy was more bottom-up than top-down, since it was based on many individual projects. Strategies include niche-specific platforms with little sharing of subsystems and manufacturing processes, horizontal leverage of key platform subsystems and manufacturing leverage, vertical scaling of key platform subsystems between high cost/high performance, mid-range and low cost/low performance levels in different market segments, and the beachhead strategy of simultaneous horizontal leverage and vertical scaling. None of these kinds of strategies could be identified in my cases, since the core platform strategy had clearly not emerged enough.

Sanderson and Uzumeri [1997] is another recent example of research on product platforms. The authors emphasise identification of design information common to a variety of product models or successive changes in their functionality, and maintaining it at a sufficiently abstract level, to keep product revisions easy and cost-effective. This is apparent in their definition of a product family as “a set of models that a given manufacturer makes and considers to be related”, not as physical products. They also base the evolution of a product family on model changes that have two components, rate (slow/rapid) and variety (one/many). Using these components, a model’s life-cycle is simple, variety-intense, change-intense, or dynamic, depending on how many changes take place and how often. Based on changes of models, the life cycle of a product family may be simple, diverse, generational, or turbulent.

From the viewpoint of customer needs, the dynamic change situation means that customer needs are diverse and change rapidly. Some customers are knowledgeable and others are confused. Competition is tangled, but intense. Companies try to respond by providing flexibility based on a balance between core competence and new products. Marketing aims at anticipating pending shifts in customer needs. A good example of this kind of situation is the end of the biggest code generation research project in 1992. VTT faced competition from a commercial tool vendor company that was its former collaborator, in trying to balance its activities on maintaining the existing competence, and to create new code generation-related innovations.

In the change-intensive case, customers are technically knowledgeable and desire only the latest and best designs. They have minimal loyalty and competitors constantly try to make current offerings obsolete. Organisational learning focuses on certain groups that try to make use of knowledge for the needs of the next product generation. Marketing concentrates on selling products and guessing what competitors will do.

In the variety-intensive case, marketing looks for product variants, and many such variants may be designed simultaneously, companies are aiming at organising manufacturing effectively. Organisational learning occurs mainly in groups dealing with certain product variants. In the case of a commodity, changes are slow and economies of scale dominate manufacturing. Organisational learning does not play any central role in this matter. This phase has not yet been reached in my two cases, and would require the development of sufficiently large specialised markets. According to Sanderson and Uzumeri [1997], from the suppliers' viewpoint the situation is as follows. In the dynamic case there are many suppliers, but they enter and exit industries relatively often. Supplier certification requires plenty of time and because technologies are changing rapidly, suppliers should possess strong technical expertise. Based on the experience from the code generation case, the number of suppliers may be considerably smaller if the market is restricted, but the other characteristics certainly hold true.

In the change-intensive case, there are few suppliers, but they are very competitive, constantly offering new and improved versions of a few designs. Customers change suppliers frequently. In the variety-intensive case, there are many stable suppliers, each of which focuses on a certain offering. A number of different technologies can be used to provide the same functionality. The current situation regarding fault diagnosis systems resembles the variety-intensive case from the viewpoint of VTT, although the market has not yet been developed correspondingly. There are only a few suppliers. However, the suppliers are rather stable. VTT has been able to focus on a basic offering whose functionality can be implemented by several different technologies, depending on customer requirements.

According to Sanderson and Uzumeri [1997], model competition leads to product family competition that can be well illustrated by the increasing variety of portable computers between 1981–1991. Companies forced to leave the market misread the technological forces that drove the variety and change of the portable computer product families. Sony's case, also discussed by the authors, was more successful.

The combination of product novelty and longevity explains the company's dominance in Walkman type products, based on the introduction of several "new model platforms" and many generational changes. Generational innovations were designed by the so-called "special projects" platform design teams. Non-platform projects for designing derivative products addressed both topological ("cosmetic redesigns") and incremental changes, and constituted up to 99% of all development projects.

This ratio of innovations and derivative products is considerably smaller than the one intended at VTT, where over a half of all projects (i.e., green and blue projects) are meant to be used to create competence and develop it further. However, for VTT it is certainly true that "little attention has been paid ... to the opportunity to evolve the management of consecutive design projects into the development of a product family" [Sanderson and Uzumeri 1997]. This is obvious from the two cases and the discussion of strategic competence management in Chapter 9.

3.2 RESEARCH INTO INDUSTRIAL NETWORKS

Various views have been suggested in networks of industrial buyer-seller relationships (cf. Easton [1992]). I will discuss the rapidly growing literature on such relationships to the extent relevant to my research, and return to the subject when introducing the competence evolution framework in the second and third parts of the dissertation. Möller [1994] is a good meta-theoretical review of network approaches, which may be viewed as an alternative to the so called marketing management research. According to Möller and Wilson [1995a], atomic markets, singular independent business transactions, and stimulus-response relationships between buyers and sellers characterise the marketing management school of thought, and Kotler is one of the best known representatives of marketing management researchers.

Möller has summarised the marketing traditions of the nineties in an unpublished piece of work, from which Figure 6 has been taken. As shown in the figure, he considers the “market strategy” to involve the resource, competence and learning view of the firm, whereas the “interaction approach” includes inter-organisational and “extensive” relationship marketing. In my research the issues become intertwined. However, my research proposition of competence evolution in project relationships is in line with the “multiple modes of marketing” and “integrated marketing” concepts that Möller likens to the interaction approach.

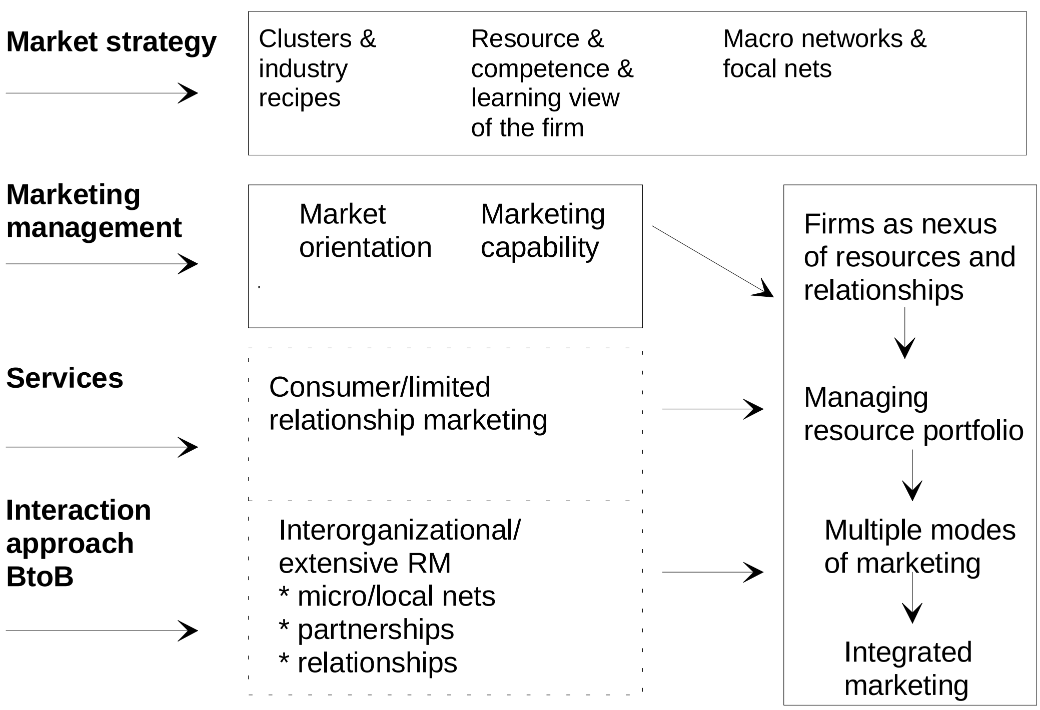

Figure 6. A view to marketing traditions 1990–2000 [Möller].

Early research on inter-organisational relationships has relied much on the social exchange theory [Thibaut and Kelley 1959]. Researchers have highlighted buyers' and sellers' motivational investments in the relationship, and their perception of the developing expectations, trust, satisfaction, and commitment [Anderson and Narus 1984, 1990, Dwyer et al. 1987, Wilson and Mummalaneni 1986]. From my point of view, this research stream does not largely involve the competence evolution perspective associated with the exchange, which I consider to be an important result of the development of inter-organisational relationships.

Researchers examining the governance structures of dyadic relationships provide another view of inter-organisational relationships. Based primarily on transaction cost economics [Williamson 1985] and on the organisational dependence theory [Pfeffer and Salancik 1978], researchers have shown how exchange conditions influence the nature of the exchange. The continuum of changes in conditions ranges from competition-dominated transactional exchange through co-operative exchange to the dominance of either party [Anderson and Coughlan 1987, Heide and George 1988].

While providing a contingency-type understanding of dependence of the relationship, governance structure research is rather static and the factors related to mutual interaction remain faceless macro forces. This line of thinking is also apparent in [Halinen 1994], which addresses a dyadic relationship between a producer and buyer of professional services, as I pointed out above. In other words, the study focuses on the outer context.

Within the Industrial Marketing and Purchasing (IMP) group and based primarily on the resource inter-dependency notion, [Campbell 1985] and [Möller and Wilson 1988, 1995b,c] subscribe to the contingency of business relationships. They argue that the mode of a relationship depends on a complex set of variables, including product-related, actor-related, organisational and industry-related factors. Compared with the rather deterministic transaction cost analysis inspired work, these authors assume a more enacted view of the business environment, incorporating experience-based differences in the environment. However, the contingency perspective is quite static and does not highlight the business dyad beyond the dyad's type or state. Competence is not addressed in any depth in this stream of research. For example, Halinen [1994] sees competence much as self-evident for a professional service firm.

A well-known conceptualisation of the evolution of industrial relationships is presented in [Ford et al. 1986]. In the classic article “How do companies interact”, Ford and others argue that an inter-company relationship is basically ambiguous rather than clear-cut. They propose four interrelated factors for characterising relationship changes: capability, mutuality, particularity, and inconsistency. *Capability* describes the relationship between the interacting parties, be they organisations or individuals, in terms of what they can do for each other by using their resources. I have taken this view to capability in my research, as will be explained in Part II. In the case of the monopoly of a supplier or a monopsonistic buyer, either party bases the relationship on a single capability, while only one organisation possesses the capability.

In perfect competition, the relationship is also based on a single capability, but that capability is a commodity. If broader resources are required in relationships, they tend to move in a direction where the differences between alternative actors are greater. This may lead to relationships based on broad resources and small differences between actors, which is rather common in industrial networks.

Mutuality is a measure of the degree to which an organisation is prepared to give up its goals or intentions in order to increase its own ultimate well-being. This factor involves a trade-off between short-term opportunism and long-term gains. Mutuality can only be demonstrated over time and critical situations – the existence of conflicts presumes a certain level of mutuality. It is common for industrial organisations to have an overall idea of mutual interests, while simultaneously being in conflict over what should be their respective contributions towards the joint achievement.

Particularity characterises the relationship in terms of its direction and uniqueness. In extreme cases, where the relationship between the parties is unique and directed solely towards each other, particularity is very high.

Inconsistency refers to ambiguity. The relationship may be inconsistent over time, or there may be inconsistency between different relationships of the same partners undertaken by different organisational units or persons. This concept focuses not only on relationships between parties with conflicting interests, but also between parties having common interests.

Inconsistency implies the opportunity for short-term expediency or changes in individual activities (cf. the *project cycle* in [Alajoutsijärvi 1996]), without changes in principal relationship policies (cf. the *long cycle* in [Alajoutsijärvi 1996]). In such a way, inconsistency captures the dynamic nature of relationships and characterises the activities of an organisation with respect to the other three factors. I have applied these four factors in my research, and will discuss them more thoroughly in the second and third parts, where the competence evolution framework is presented and revised.

As I understand, one of the main differences between various concepts of relationships is related to how [Ford et al. 1986] considers inconsistency, and could be called the transactional vs. the relational viewpoint. The latter assumes that no single transaction can be understood without considering how it relates both to past transactions and the “surroundings” of ongoing transactions. This also became clear in my own research, if individual projects are viewed as the main type of transactions.

Möller and Wilson [1995b] present a taxonomy of buyer-seller interaction factors including the environmental context, supplier and buyer characteristics, task characteristics, and outcome characteristics. The exchange process consists of the exchange of physical and immaterial resources and social resources. Exchange processes can be captured by describing them as episodes. Monitoring of performance and conflict resolution are typical co-ordination process tasks. The authors list as many as thirty-five organisational characteristics affecting relationships.

The outcome factors include states of interacting factors and processes, as well as changes in the environmental context. The taxonomy of interaction factors is considered as “useful mainly for presentational purposes”. I found the notion interesting, and tried to apply it at an early phase of my research. However, I learnt that one of the basic difficulties of generic taxonomies is their abstract nature. Instead of using tens of factors I decided, after some time and struggle, to choose only three main concepts: actors, resources, and activities. Obviously I then needed to extend these concepts towards the kinds of factors proposed by Möller and Wilson. However, the fixing of the few core concepts facilitated the matter considerably. For finding the factors, I relied heavily on [Alajoutsijärvi 1996], not only because the author was the supervisor of my doctoral studies, but also because the research carried the idea of studying reciprocal network phenomena further by analysing successive episodes consisting of development projects.

Alajoutsijärvi focuses on very long-term relationships, sometimes over forty years, within the context of customised investment goods. The seller he studies is a paper machine manufacturer and its customers are a few paper mills. A dyadic relationship model is developed and evaluated by Alajoutsijärvi in two phases, along the lines of Halinen [1994]. His a priori model is based on literature, the model is revised on the basis of a study of three cases. The model includes four states of relationships, co-operation, customer’s dominance, producer’s dominance, and competition. I have utilised the same states as a part of my competence evolution framework.

One of Alajoutsijärvi’s key ideas is to view the relationship states as a continuum, so that at some point the state may include aspects of all four states. The state space is illustrated as a diamond-shaped surface, known as a “sal ammoniac” candy in Finnish. A state may be seen as an attribute of a relationship. It can have one of the four extreme values or some other value in between. In order to recognise the value, the state of the relationship must be defined operationally.

Conditions causing changes of a state include the results of the relationship, the parties involved, the local network – which is called the focal net in my research – and the macro forces of the environment. Combinations of these conditions are not much discussed by Alajoutsijärvi, which means that he does not view the conditions an integrated system. In a longitudinal study involving intangible resources this kind of a system would, indeed, be difficult to create and maintain. Therefore in this research, I only provide a rough characterisation of the nature of relationships during certain periods of time, with an analysis of the main conditions having affected them.

On the basis of [Möller and Wilson 1995b], Alajoutsijärvi considers relationships to involve technical-economical and psycho-social outcomes on the customer and producer sides, both at group and personal levels. I address both types of outcome as a part of the competence evolution framework, but mainly at the level of research groups and project teams. Alajoutsijärvi investigates how the outcome of a successful or an unsuccessful project affects the relationship.

Since the period of time that I investigated was fifteen years and most of the individual project relationships lasted only for a few years, it was difficult to record such changes by any other means than the historical value of a relationship, as seen by different parties. On the other hand, successful and unsuccessful projects certainly affected the continuity and type of the relationships that I was studying.

The forces that Alajoutsijärvi lists as affecting the success of a single project include forces in each phase of the project, crises and personal conflicts, the project contract and plan, and the effects of top management of the parties. Nevertheless, Alajoutsijärvi does not classify successful or unsuccessful projects in terms of the solutions that were provided by the seller to the buyer, i.e. the effects of the machine manufacturer's competence in providing paper mills to its customers. In my research, the success of a relationship in terms of continuing projects depends first of all on the capability of the seller to actually create and deliver the competence bought by the customer, which can not be taken for granted.

According to Alajoutsijärvi, the characteristics of the parties affecting the state of a relationship include the relative size of each party, the preferred interaction style, the familiarity of the partners, and the centralisation of the purchasing function of the customer. Again, the classification takes the technical capability of the machine manufacturer pretty much for granted and vice versa, it presupposes the buyer's ability to specify the requirements of a new or an improved paper machine in a successful manner. In Alajoutsijärvi's research, the mutual roles of a machine manufacturer and paper mills are correspondingly seen as quite distinct. In this kind of setting, the competition faced by the seller in a relationship is directed towards third parties, not towards the buyer's own resources. In my research, the competitive advantage of the buyer himself is often quite considerable.

In order for a relationship to change, the parties have to behave accordingly. According to Alajoutsijärvi, the behavioural modes on which changes of relationships are based include the following: number of alternative parties, construction of the relationship infrastructure, transferability of behavioural styles, mutual rules for interaction, expectations concerning the relationship, and the use of power. However, Alajoutsijärvi does not address the roles of certain behavioural modes in certain changes. In my research context, the basic atmosphere of a project relationship depends mainly on the economic setting; as the techno-economical outcome of the relationship, the customer may pay less than one half or the full market price for the service provided.

The relationship factors Alajoutsijärvi proposes are motivation to invest in the relationship, particularity, mutuality, inconsistency, experience of the other party, distance, commitment, dependence, exchange processes and business communication, interaction style, evolving relational infra-structure, and connectivity to alternative partners. I have chosen a simpler set of factors based on [Ford et al. 1986], including three of these factors: particularity, mutuality, and inconsistency, in addition to capability.

The directions of further research Alajoutsijärvi points out include aspects of "long cycles" of relationships, and "project cycles" of which long cycles consist, as well as a study of the project typology made explicit in the analysis of the case example. These are all addressed in my research, even though the long cycles I studied were, as a matter of fact, quite short.

According to Håkansson and Snehota [1995], using the activity-resource-actor (ARA) model is based on three layers of the "substance" of a business relationship: activities, resources, and actors. The effects of a relationship based on the three layers are called functions. They can be identified for an individual company, for a dyadic relationship, and for a network of several actors. Activity structures, resource collections, and organisational structures can be formed at company level. At the relationship level, there are activity links, resource ties, and actor bonds. Networks consist of activity patterns, resource constellations, and webs of actors. From the viewpoint of company resource development, the main problem is how to balance these functions in evolving relationships and networks.

Håkansson and Snehota claim that the type and strength of activity links are critical for the success of a relationship, that resources are brought together, confronted, and combined in resource ties, and that commitment, identity, and trust affect actor bonds. They identify normative, managerial implications of the effects of relationships in three areas: capability development, marketing and purchasing, and strategy development. The main management task is "to keep the customer and supplier relationships productive", with which, on the basis of this research, I absolutely agree.

Håkansson and Snehota state that the activity dimension creating and making use of resources is not easy to treat analytically. However, I have addressed the issue, not to mention Dubois [1994], whose dissertation is titled to "Organising industrial activities – An analytical framework". Although Håkansson and Snehota claim that "to what extent the design of the activity structure [of a company] has been conscious can be debated", they consider the following managerial questions concerning activity links: How to develop and handle activity links in a single relationship? How to simultaneously exploit the set of the company's relationships and activity links?, and How to develop the company's position in the overall activity pattern? Two other questions arise from the viewpoint of capability development: How to relate relationships systematically to the activity structure?, and How to combine or connect relationships with each other? All these questions are relevant from the strategic competence management perspective that is discussed in the fourth part of this dissertation.

Håkansson and Snehota argue that it is possible to study forces which cause relationship changes, and that an understanding of the mechanisms and processes of change is more important than an effort of trying to foresee them. "To put it somewhat awkwardly, it seems useful to analyse the effects after the fact rather than before". They suggest three pairs of change vectors for relationships: structure – make heterogeneous, specialise – generalise, and hierarchize – heterarchize. In this research I have used the portfolio of green, blue, and red project relationships instead, since it implements the basic relationship strategy of the focal organisation.

[Halinen 1994] is one of the few relationship studies carried out within the context of professional services. Halinen claims that "as far as service marketing is concerned, there is only a handful of studies that even consider the dynamics of exchange relationships (...) no attempt has been made to describe the development of exchange relationships conceptually". The analytical framework she uses for studying process phenomena includes the viewpoints of context (why), content (what), and process (how). Halinen addresses all these viewpoints in the relationship of an advertisement agency and its client. What concepts can be used to describe the relationship, what processes change it, and which contextual factors affect the changes? In my research, the views of interacting parties are also studied with respect to relationship changes, not only to contextual factors.

The customers of an advertising agency "purchase knowledge and creativity embodied in individual people". In my research, the situation is similar, but at the same time completely different. The customers of VTT purchase knowledge, skills, and competence embodied and shared by individuals, but controlled by the organisation. The exchange processes that are the core of interaction in both Halinen's and my own work include planning, production and delivery of services. However, contractual R&D projects are usually conducted more formally than the exchange processes studied by Halinen.

The perceived outcome of a relationship in Halinen's research includes performance of the exchange process, performance of the exchange relationship, and a psycho-social outcome of the exchange relationship. Instead of this, I address product-related, process-related, and organisational competence as the most important types of outcome of relationships – which is, of course, the question of what viewpoint is taken to relationships. I have also studied interaction between groups of actors, in order to analyse views to psycho-social outcomes of relationships.

Halinen identifies three recurring developmental phases of relationships that she calls cycles of development: cycle of growth, cycle of decline, and cycle of maintenance. I am more interested in the change of competence, which depends on the development of project relationships. The phase of the whole portfolio of project relationships involving certain competence is important in this regard, i.e. the set of green, blue and red projects.

This is related to the fact that the "width" of the portfolio of project relationships is important, not necessarily the length of customer relationships based on successive projects. The reason is that width is, in practice, needed for developing core competence. Interestingly, this is also shown in [Ford et al. 1998], on the basis of a study carried out among 123 Swedish companies. Technological relationships with horizontal units, e.g. external research partners, were the shortest when compared to relationships with customers and vertical suppliers. In 55% of the cases surveyed, their duration was less than four years. The weighted average of technological relationships with horizontal units was eight years. Ford and the others conclude that with respect to technological relationships, the most common type of companies were "broad co-operators"; 48 of the 123 companies had a versatile portfolio of relationships with customers, suppliers, and horizontal units. Thirteen of those 28 "focused companies" interested only in certain types of partners, dealt with horizontal units.

3.3 SUMMARY

In the first part of the dissertation I have outlined the purpose and context of my study, as well as defined the research problem and the approach I chose in order to solve the problem. Some of the difficulties I encountered in defining the problem and developing the research design may be related to the fact that network researchers have most often not investigated resources and activities in many details (exceptions do exist, e.g. Anderson [1994] regarding resources and Dubois [1994] regarding activities). Vice versa, the resource-based view to firms seems to have followed mostly the intra-organisational rather than inter-organisational approach. For example, Easton and Araujo [1996], who address firm resources explicitly in the context of external relationships, can be considered rather as an exception than a rule in [Sanchez et al. 1996a].

Fortunately, however, one of the best known models of networks of industrial relationships – the ARA model [Håkansson and Snehota 1995], includes activities and resources as its key concepts. Following the idea made explicit, for example, by Rosenbröijer [1998], I decided to consider competence as the creation and use of resources by relationship actors. This view seemed to fit well with the context of my research, since VTT as a contract research organisation offers services based on intangible and physical resources it creates and makes available to customers as part of R&D. These activities are almost invariably conducted in close co-operation with customers, usually in projects with both parties working together.

I therefore started to build a framework to investigate competence evolution in R&D projects based on the ARA model, as will be discussed in the next chapter. The original idea was simply to associate the elements of contractual project-based R&D services to activity, resource and actor concepts. However, this was only the beginning. Although Håkansson and Snehota [1995], for example, address resource transformation and transfer types of activities, the ARA model does not as a framework provide much support to make explicit how activities, resources and actors may change over time. This is well illustrated by Anderson [1994] and Dubois [1994], who devote their dissertations to show and analyse changes of resources and activities, respectively. I also needed to develop concepts for understanding changes in R&D projects.

The change-driven view to both competence and relationships became even more important when I addressed the code generation case. I had to revise the initial framework, not only by associating changes of activities and resources more tightly together by using process-related concepts, but also by showing how the objectives and logic of action of groups of actors changed over time.

I will now present the initial competence evolution framework, built based on the ARA model, and use it then to analyse the fault diagnosis case.

PART II: SUBSTANCE OF COMPETENCE

The second part of the dissertation focuses on competence in the context of engineering research and development. An initial version of a framework to understand the substance of competence is presented, based not only on my pre-understanding formed during the period of over sixteen years spent in conducting contract R&D at VTT, but also on an analysis of literature of both competence and networks of industrial relationships. The framework is used to investigate the fault diagnosis case. This part of the dissertation is in part based on [Seppänen et al. 1998a,b], although certain matters are not discussed in as many details as in the two referred publications.

4 INITIAL FRAMEWORK

The competence evolution framework presented in this chapter consists of two layers. One of the layers used to explain the basic elements in contractual R&D is called the *substance layer* of competence (cf. [Håkansson and Snehota 1995]). The other is called the *dynamic layer*, and it describes changes of the elements of the substance layer over time.

4.1 SUBSTANCE LAYER OF THE FRAMEWORK

The substance layer of the competence evolution framework is based on the ARA model (cf. [Håkansson and Snehota 1995] for a detailed discussion of the model), Table 2. It describes activities carried out to create and make use of R&D resources internally and together with external parties. The framework includes a typology of competence concepts, developed by extending and reformulating the resource typology given in [Rosenbröijer 1998], and by associating it with a typology of activities. Firms, relationships and networks form the interaction dimension of the substance layer. The layer also addresses the main characteristics (cf. attributes in [Easton and Araujo 1996]) of the concepts.

Typologies are used to define classes and types of concepts and values to define their attributes. The existence of certain types of concepts and the values of their attributes depend on time, which is also made explicit. Rosenbröijer [1998] uses a resource typology based on financial, physical, organisational, human, technological and reputation resources. I have modified the typology and associated it with another typology of activities, actors and relationships. These typologies are by no means comprehensive, still they were sufficient for understanding and explaining the fault diagnosis case that I studied by using the initial framework. The classes, types and attributes of the concepts form a hierarchy, a matter quite critical for ensuring the usability of this kind of a conceptual framework.

Table 2. The substance layer of the framework (with examples).

INTERACTION/ SUBSTANCE	**FIRM**	**RELATION-SHIPS**	**NETWORKS**
COMPETENCE			
Activities	*Activity structures:* Internal pre-study done based on literature	*Activity links:* Contractual development conducted for one customer	*Activity patterns:* Joint research carried out in a national research program
Resources	*Resource collections:* Use of embedded systems modelling techniques and tools in the development of fault diagnosis systems	*Resource ties:* Creation of fault diagnosis features for an automation system built by a customer company	*Resource constellations:* Knowledge acquired by VTT's and a customer's experts from the customer's application engineers
ACTORS			
Parties	*Organisational structures:* Internal quality review board	*Actor bonds:* Project management group	*Actor webs:* Joint project team of VTT and its research partners

Table 3 summarises the activity and resource concepts of the framework as they have been used in this research. The attributes included in [Seppänen et al. 1998a] will only be mentioned here. Their practical usefulness was not obvious, as will be discussed later.

The engineering R&D service, for example software design, implementation and testing, is the main *technological resource* utilised in project relationships. Possession of some knowledge, skills and capabilities, i.e. *human resources*, are required from individuals to be able to carry out the service. *Physical resources* are simply considered to be either products, documents or system development tools. Human, technological and physical resources become often tightly intertwined in R&D projects. Even though *Temporal resources* are crucial for project-based research and development they are missing from the typology used in [Rosenbröijer 1998]. The basic types of temporal resources planned for and controlled in projects, schedules, efforts and calendar time, are therefore included in Table 3. The project management system is one of the most important *organisational resources* of VTT, reputation is used e.g. in marketing. *Financial resources* are usually exchanged in contractual research and development, be it internal or involve external parties.

Table 3. Typology of R&D resource and activity concepts.

Elements	Classes	Examples of types
Resource		
	Human Technological Physical Temporal Financial Organisational Reputation	Knowledge, Skills, Capability Engineering R&D service Product, Document, Tool Calendar time, Schedule, Effort Expense, Income Project management system Professional reputation
Activity		
	Human	Learning, Doing, Management, Evaluation
	Technological	Tracking, Acquisition, Planning, Use, Transfer, Integration, Evaluation
	Physical	Tracking, Acquisition, Use, Transfer, Integration, Evaluation
	Temporal	Planning, Use, Evaluation
	Financial	Planning, Use, Evaluation
	Organisational	Project management
	Reputation	Professional activities in the community

As shown in the table, activities are needed for acquiring resources, planning for, carrying out, evaluating and utilising the results of research and development, supporting individuals in developing and extending their expertise, taking care of project management, planning and controlling financial matters, following general technical developments, acting as a member of the professional community, etc. In [Seppänen et al. 1998a], different attributes were used to indicate the characteristics of competence (cf. [Easton and Araujo 1996]). I will focus on one characteristic of technical competence; the content of research and development services. It is viewed, as one of the original contributions of this research, through four dimensions derived from the strategic market model of Abell [1980]:

- the *application* domain and particular products involved,
- the *functions* accomplished by products,
- the *techniques* on which the functions are based, and
- the implementation *technologies* used to realise products.

The dimensions become intertwined in R&D services and products. Therefore, they must be defined for both types of resource concepts. “Maturity” can be used to describe the codification of the concepts. For physical resources of the types “Document” and “Tool”, the attribute “Existence” can be defined instead, because their form of appearance may not be interesting [Seppänen et al. 1998a].

In the fault diagnosis case to be analysed in Chapter 5, the functional dimension of R&D service and product resources includes the so called generic fault diagnosis tasks, around which the functionality of fault diagnosis systems was developed. The fault diagnosis applications developed by VTT range from machine automation to telecommunication, involving different types of products, such as elevators, rock drilling machines and GSM network equipment. Several software and knowledge engineering techniques have been used to design the systems and various technologies applied in their implementation (cf. [Kurki 1995]).

Elfring and Baven [1996] separate functional capabilities from application capabilities, as discussed in Chapter 3, which fits with two of the dimensions listed above. Abell [1980] suggests three of the dimensions, i.e. applications, functions and technologies, both as a means for structuring markets and choosing the competitive foci of a firm.

However, both authors miss the “techniques” dimension, which is central in engineering research and development. Another well-known term that could be used instead of "technique" is "discipline". The whole engineering education is based on different disciplines, which illustrates the importance of this dimension. The basic technique needed in the two cases of this research is software engineering. In the fault diagnosis case one of its sub-disciplines, knowledge engineering, was heavily utilised. This technique was actually the starting point of the fault diagnosis research in the focal organisation. I will use the four dimensions to analyse how implicit individual skills reverse to explicit organisational competence, which may ultimately appear in such tangible form as a fault diagnosis platform of VTT or the fault diagnosis system of the paper winder of a paper machine.

A typology of actors and relationships for the needs of this research is shown in Table 4. Industrial segments are clusters of organisations active in certain application domains, such as machine automation and telecommunication. Research institutes, universities, public funding bodies, government authorities and companies are typical brands of firms involved in research and development. Tool vendors are a special type of companies. Organisational units include both permanent and temporal groups, such as departments and project teams. To put it simply, the main types of persons involved in research and development are managers, researchers, product developers and end users.

Inter-actor relationships between firms are often heavily project-based in contract R&D. Projects could also be considered as a special type of actors. Nevertheless, in this research I wish to study individual project parties. In addition to projects, special business transactions and broad professional collaboration schemes exist. The former are events taking place in a short period of time compared to projects – consider purchasing of equipment or invoicing as examples. The latter are needed for people and organisations to co-operate in professional communities. Relationships can be characterised, for example, by attributes like “volume”, “type” (c.f. green, blue and red in the case of VTT) and “state” (c.f. competitive, co-operative, dominating, submissive in [Alajoutsijärvi 1996]).

Table 4. Typology of actors and relationships involved in R&D.

ELEMENT	Classes	Examples of types
ACTORS		
R&D party	Industry Firm Organisational unit Individual	Industrial segment Research institute, University, Funding body, Authority, Company, Tool vendor Permanent group, Temporal group Researcher, Product developer, Manager, End-user
RELATIONSHIPS		
R&D relationship	Transaction Firm Relationship Network Environmental	Meeting, Purchase, Invoicing Internal green project External dyadic or joint red project External joint blue project Professional collaboration

4.2 DYNAMIC LAYER OF THE FRAMEWORK

The dynamic layer of the framework makes explicit how and why the concepts of the substance layer change over time. The substance layer is based on two dimensions: substance concepts consisting of resources, activities and actors, and the interaction of actors. By using the dynamic layer, I will approach the question of how the evolution of competence, activities carried out by actors to develop and use resources, is affected by the relationships of the focal organisation. Therefore, I will start the discussion of the dynamic layer from changes among project relationships.

4.2.1 Changes among relationships

I will address changes among projects rather than in the other types of relationships shown in Table 4. The basic kinds of changes that may occur in project relationships are rather simple, as shown in Figure 7: establishment, dissolution or continuation of a project (“start, finish, continue”), changing of a relationship to a network (“expand”), and vice versa (“specialise”), and changing of an internal project to external (“exploit”), or vice versa (“research”). Figure 1 indicates how the number and role of external parties usually varies in these changes. Although, in principle, each type of a project relationship may change into any other type, the strategic goal of VTT is first to extend its internal self-funded green activities to joint blue projects, and then to specialise them on several red contractual relationships, or if possible, to expand them to red networks. Shortcuts to red contractual projects from internal green activities are possible, although risky, whereas reverse changes are more rare.

The finishing of internal activities without any continuing external relationships is usually considered as a failure, whereas the possible negative effects of the ending a relationship or dissolving a network depend on the financial and technical resources involved. A relationship may continue, sometimes for several years, in the form of successive projects.

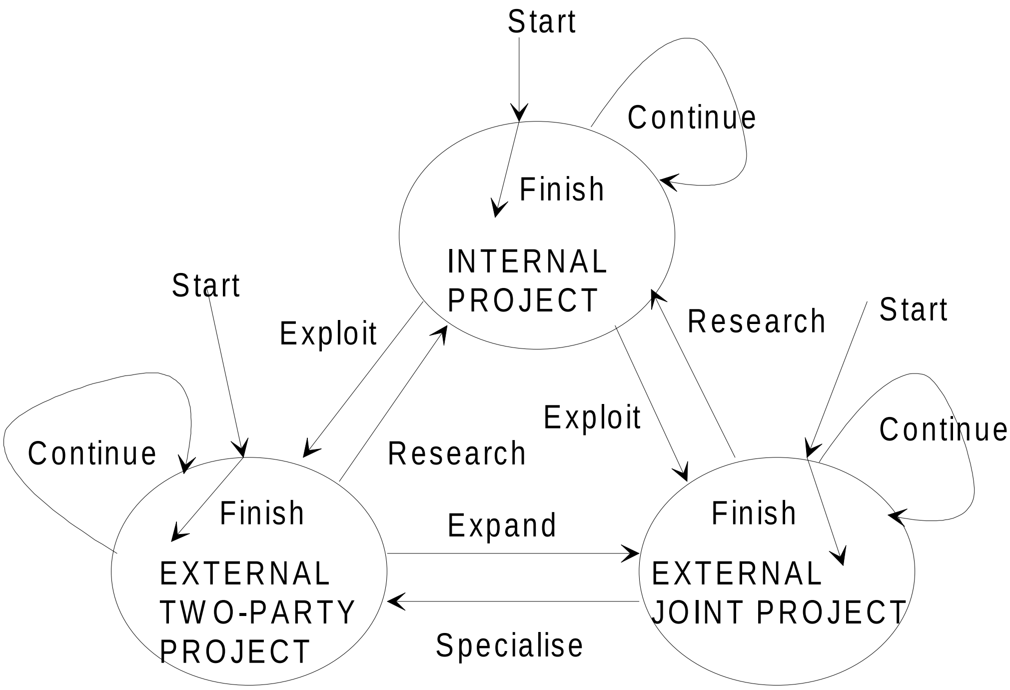

Figure 7. Changes among project relationships.

The changes are affected not only by the participating actors, but also by broader macro forces (cf. [Alajoutsijärvi 1996]). For example, the readiness of firms and industries to use certain technologies, produce macro forces that affect relationships (cf. [Uusitalo 1995]). I will discuss the effects of macro forces in the context of this research in Section 7.2.

4.2.2 Changes within relationships

Since most projects involve external actors, one of the main relationship management tasks is the control of resource exchange and sharing between the participating actors. In contractual research and development the source of financial resources is usually explicitly defined, whereas the ownership of technical resources may be a much more controversial issue. In the case of VTT, the portfolio of green, blue and red projects based on different sources of financial resources affects the ownership of technical resources acquired or developed in connection with R&D activities. Changes between and within the firm, relationship and network levels must therefore be analysed in terms of the change of colours of projects. For example, networks may exist both as joint red customer consortium projects and joint blue research projects, while the different sources of financial resources make the networks completely different as far as the roles of participating actors and the ownership of technical resources are concerned.

These phenomena affect relationships in a complex manner over time. I will use the four relationship attributes discussed in [Ford et al. 1986], i.e. capability, mutuality, particularity and inconsistency, to analyse changes in relationships. Table 5 shows examples of the attributes, and how they affect each other. Viewed from the perspective of the ARA model, particularity and inconsistency involve actor bonds and webs in external relationships and networks, capability concerns resource ties and constellations, and mutuality deals with activity links and patterns.

Table 5. Interrelationships between the relationship attributes.

Attributes	Particularity *(Actor bonds, webs)*	Inconsistency *(Actor bonds, webs)*
Capability *(Resource ties, cons-tellations)*	[1] Example: specialisation in a certain customer application	[3] Example: only a few persons possess knowledge of a technology important to many customers
Mutuality *(Activity links, patterns)*	[2] Example: a contractual consortium concerning a new technology or an emerging engineering technique	[4] Example: inexperience in the development of a customer's application

Four change management functions involving the six basic ARA relationship functions shown in Table 5 can be identified on the basis of the interrelationships (cf. [Alajoutsijärvi and Tikkanen 1999a]):

- [1] Balancing the particularity of resources (capability vs. particularity)
- [2] Balancing the mutuality of activities (mutuality vs. particularity)
- [3] Assuring the desired level of capability (capability vs. inconsistency)
- [4] Assuring the desired level of mutuality (mutuality vs. inconsistency).

"Balancing of the particularity of resources" concerns the question about the extent to which an actor should tailor its resources towards relationships with particular parties, i.e. how to deal with resource ties and constellations. The particularity of an organisation's resources has long-term strategic implications because of the possible non-transferability of investments made in certain relationships. Highly relationship-specific resources, e.g. skills serving only one customer, may prove hazardous in the long run. An example would be specialisation in the contractual development of the applications of one big customer only. Overly general resources, such as knowledge available from software engineering textbooks, may, on the other hand, not facilitate establishing and maintaining specific relationships. In the fault diagnosis case avoiding of this kind of a situation paved the way for the focal organisation to create and deploy competence in fault diagnosis of automation applications. Some early competitors lagged behind, because they wished to continue building expertise in generic knowledge engineering techniques used in fault diagnosis systems.

Because of the risks involved in the lack or excess of particularity, actors must be concerned about the extent to which the mutuality of their activities is related to the particularity of their resources. The corresponding change management function involving activity links and patterns is called the *"balancing of the mutuality of activities"*. For example, if the level of mutuality within a particular contractual consortium is perceived as high, the participating organisations may decide to make expensive relationship-specific investments or hire personnel experienced in a specific common technique, to facilitate the co-operation. In the early phase of fault diagnosis research at VTT – and in Finland in more general terms, such investments were made in special design tools, as part of joint projects between industry and research institutes interested in knowledge engineering techniques.

The management of inconsistencies involves *"assuring the desired level of capability"* and *"assuring the desired level of mutuality"* in actor bonds and webs. Gaining the desired level of capability concerns, for example, the development of skills among those individuals in the organisation involved in certain relationships. Seen from the customer's perspective, there may be not enough human resources with the necessary skills available in the organisation, or the skills may vary over time. It may be wiser for a contract research organisation to have full human resource allocation rights for its relationships, in order to be able to manage capabilities as an institution. On the other hand, it is possible that customers like to be served by persons who they know personally (cf. [Halinen 1994]).

Assuring the desired level of mutuality relates to consistency in the activities of interacting persons. For example, inexperienced persons may be allocated to projects where application understanding is crucial. This is very well illustrated in the fault diagnosis case. After several years of collaboration with automation firms, VTT addressed a new application domain for fault diagnosis, telecommunication network equipment. This domain was quite new to the focal research group as a whole. Remarkable difficulties emerged, and the attempt to transfer existing competence to the customer failed in the first project. The customer consortium established for the project was dissolved, but fortunately one of the participants wished to continue the work as a contractual dyadic project. VTT invested in learning the application domain, became a strategic partner of the customer and even broadened its work to the customer's customer.

4.2.3 Managing of competence changes

As indicated above, changes in project relationships and activities carried out on resources can be related to each other via the four change management functions, by which the desired level of particularity and capability of resources and mutuality of activities are ensured and maintained. Competence changes involve yet another functions creating and changing the codification and life cycle of resources and institutionalisation of activities. I will use five such functions: problem-solving, diffusion, absorption, scanning and life-cycle evolution (cf. [Boisot 1994]).

Managing of particularity changes – problem solving and absorption

Particularity of a resource depends on codification, which can be *tacit*, *portfolio-specific* or *generic*. From the relationship perspective, the codification involves contextuality. Certain relationships result often in tacit knowledge specific to the relationship. VTT persons gained special mine related expertise from the development of fault diagnosis systems for rock drilling machines in projects carried out for a company manufacturing such machines in Finland. Portfolio-specific capabilities could, on the other hand, be reused in the development of fault diagnosis systems for other automation applications, such as hot strip steel mills and paper machines. A portfolio of a large number of relationships with automation firms emerged in the focal organisation in the mid-nineties. Certain generic knowledge engineering skills were used in the new application domain for fault diagnosis, telecommunication networks, in the late nineties.

Problem solving refers to the task of codifying and giving a structure to tacit knowledge possessed by individuals. I see problem solving as a key competence evolution function, regarding the creation of core products and platforms from tacit skills and portfolio-specific capabilities.

Absorption denotes a reversed task of applying codified knowledge to different situations in a "learning by doing" or "learning by using" fashion. For example, the use of an existing fault diagnosis platform in a new relationship, if successful, is based on the absorption of codified fault diagnosis knowledge in the context of a new customer application.

Diffusion refers to the institutionalisation of knowledge within a larger community of people. Diffusion of well-codified and generic information is less problematic than diffusion of data which is implicit and relationship-specific. This was one of the reasons for knowledge engineering related professional collaboration in Finland in the late eighties.

Scanning, a reversed function, refers to the identification of the threats and opportunities that exist but are hidden in some data. Scanning transforms the data into unique or idiosyncratic insights that come into the possession of groups or individuals. Scanning can be rapid when the data is well codified and abstract, but very slow and random if the data is implicit and relationship-specific.

Capability changes – life-cycle of resources

The content of a resource affects capability, but the effects of the content on particularity and mutuality depend on the *life cycle* of a resource – which I consider to be the change of the content of the resource over time. I will use the concept dominant design (cf. [Uusitalo 1995], who analyses this concept, originally proposed by Tushman) for identifying the life cycle. According to Uusitalo, evolution of industrial designs includes the following phases: era of incremental change, discontinuity, era of ferment, era of substitution, threshold of substitution, era of design competition and dominant design.

A simplified life cycle of the content of a resource consists of the phases *incremental changes*, *discontinuity*, *substitution*, *competition* and *dominance*. I have identified these phases for each of the four dimensions of the content of competence in the two cases I have studied.

One could claim that e.g. generic engineering disciplines are less specific for a certain relationship than some application-specific knowledge. However, this is not necessarily the case. For example, although the initial fault diagnosis relationships of VTT were highly particular, they were based on a mutual interest of highly generic knowledge engineering techniques. The techniques dominated fault diagnosis related resources at the end of the eighties. High particularity of the resources presupposed the use of these techniques, and high mutuality meant seeking contacts only with knowledge engineers, rather than also with application experts or end users.

The relationships of the focal organisation with other parties may help maintaining or changing the life-cycle phase of a specific dimension. This may affect the particularity of the resource to be exchanged or the mutuality of the activities performed in connection with the relationship. Codification and institutionalisation of a resource are also related to the life cycle. For example, if an organisation wants to exploit a common core technology, it must have access to explicitly codified information of that technology.

The *evolution of the life cycle* of a resource involves both the relevant dimensions of the content (cf. [Abell 1980]) and the life-cycle phase of these dimensions. A company may, for example, focus on one or more of the dimensions itself and take care of the other dimensions through the external relationships. Depending on the scope of the resources in question, the company may need to manage the content of several interrelated resources, so that they are combined for the strategic goals of the organisation. If the combination involves the resources of any external parties, absorption and diffusion of the content of the own resources must be controlled. A contract research organisation may possess knowledge of hundreds of techniques and technologies, deal with tens of applications and focus on many functional problems. When establishing and maintaining its relationships, the company must manage the life cycle of at least the most critical resources that it creates or exploits in certain relationships.

Since I consider competence involving resource creation and usage activities, the dynamic layer of the competence evolution framework should also address changes in R&D activities. One of the most important ways of managing activities in a contract research organisation is a project management system, which I have modelled as an organisational resource. However, the initial version of the framework presented in this chapter addresses changes of activities only through changes of resources. If, for example, the codification of a resource is increased through problem-solving, the activities of the focal organisation would change from individual learning to group-related transfer and combination of explicit technological skills. I will revise the initial framework by some process-related concepts to address changes of activities in a more rigorous manner in Chapter 6, where the framework as a whole is reconsidered to analyse the code generation case.

Figure 8 illustrates the dynamic layer of the framework. Assuring of capability by life-cycle changes, balancing and assuring of mutuality by diffusion and scanning, and balancing of particularity by problem solving and absorption are illustrated in the figure as orthogonal dimensions interrelated to each other through the four relationship attributes, and thereby to the activity, resource and actor functions of the substance layer. For example, inter-personal activity links result usually in tacit relationship-specific skills of individuals, whereas collaborative inter-organisational activity patters involve generic well-codified knowledge. Similarly, individuals are often behind resource ties that may lead to radically new capabilities, but organisations and formal collaboration networks are needed to establish resource constellations for well-established competence.

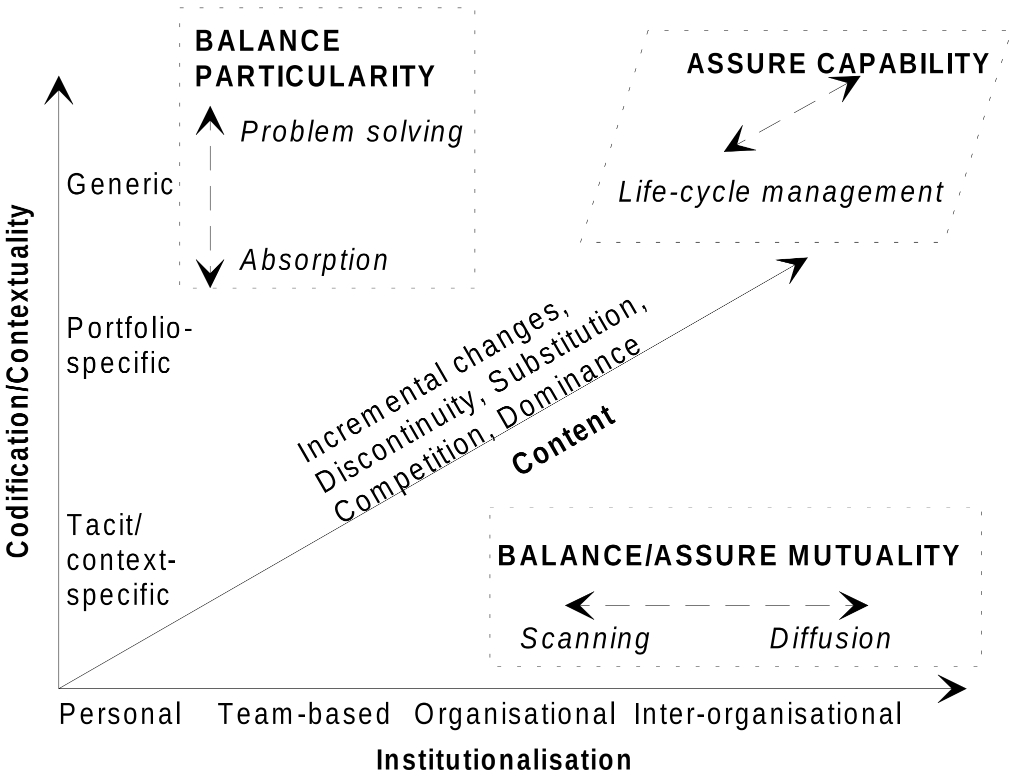

Figure 8. Concepts of the dynamic layer of the framework.

To summarise the key concepts of the framework concisely:

- *competence* was associated with activities used to create and utilise resources within the context of R&D, using the ARA model,
- *R&D projects* were likened to relationships using the ARA model, so that actors may be individuals or organisations,
- *change of R&D projects* was modelled as relationship changes, and
- *evolution of competence* in R&D projects was modelled using four relationship attributes (Table 5) and four resource and activity change management functions (Figure 8).

5 THE FAULT DIAGNOSIS CASE

Appendix 1 includes a chronological description of the fault diagnosis case, written by myself. In the following I will first discuss the substance of the fault diagnosis project relationships and competence, and then address their change, analysed using the dynamic layer of the competence evolution framework. [Seppänen et al. 1998a, b] include rather detailed discussions of the substance of the fault diagnosis case, where the case data was used to demonstrate the initial competence evolution framework. I will return to the main fault diagnosis substance elements in Chapter 8 for a cross-case analysis, using the modified framework.

5.1 THE FAULT DIAGNOSIS SUBSTANCE

Table 6 and Figure 9 illustrate the main periods and aspects of fault diagnosis research and development in the focal organisation from 1986 to 1997. Changes in fault diagnosis R&D, as they were identified from the case data and are depicted in Figure 9, will then be analysed by using the initial competence evolution framework.

Table 6. Overview of fault diagnosis R&D in between 1986 and 1997.

Period	Main resources	Activities	Actors	Relationships
1986–1988	Knowledge engineering techniques	Literature surveys, pre-studies	VTT	Green projects
1989–1991	Knowledge engineering techniques, Expert system technologies	Research of fault diagnosis features	Process and machine automation firms, VTT	Blue and red projects: interaction of knowledge engineers
1992–1994	Diagnosis functions, Automation applications	Development of fault diagnosis systems for automation applications	Process and machine automation firms, public funding bodies, VTT	Red and blue projects: interaction of diagnosis and automation experts
1995–1997	Core fault diagnosis platform, case-based reasoning, modelling techniques	Development of a core platform, development of fault diagnosis systems and prototypes	Automation and telecom firms, public funding bodies, VTT, other research institutes	Green, blue and red projects: interaction of diagnosis and application experts and end users

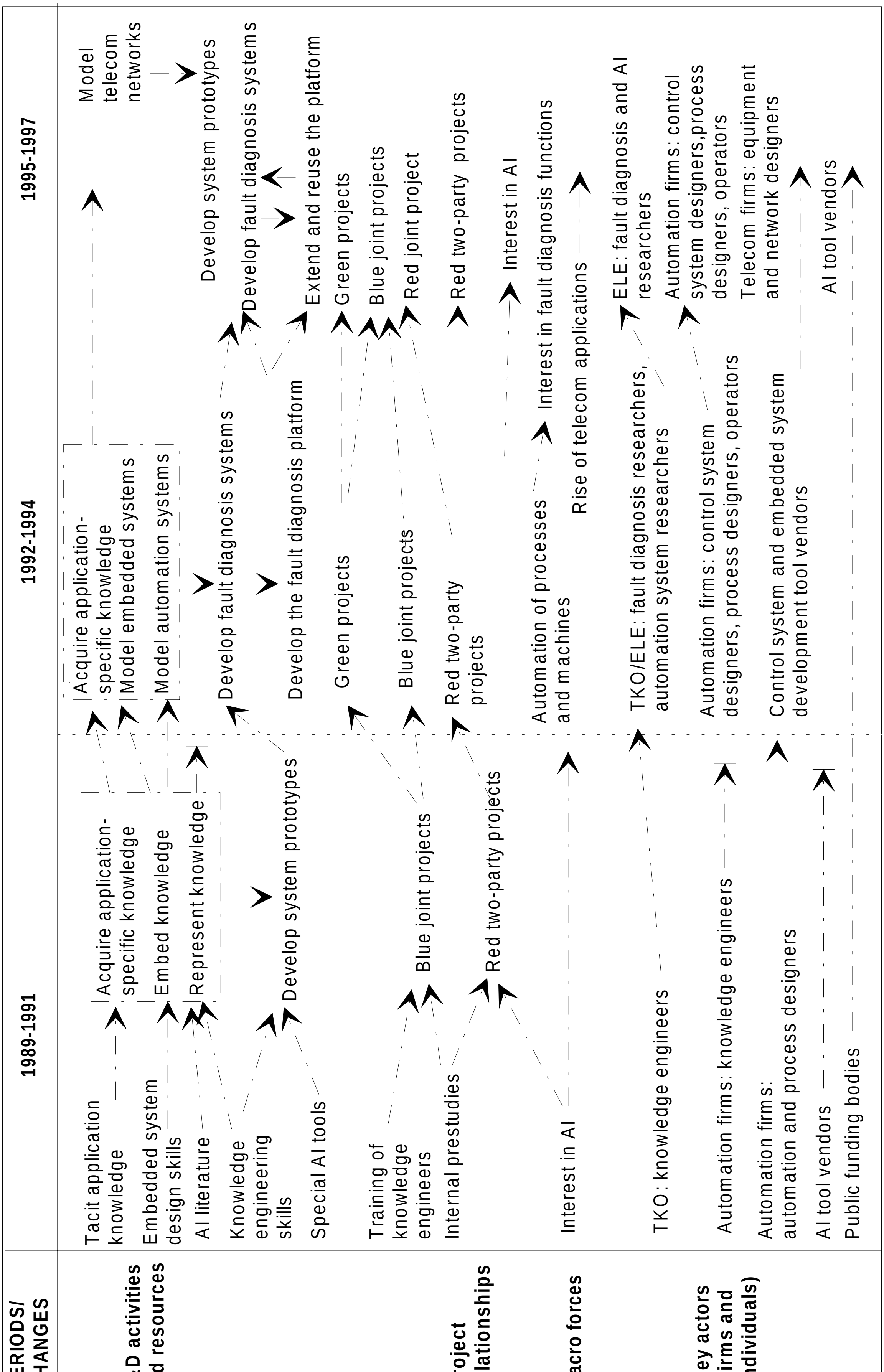

Figure 9. Changes in fault diagnosis R&D in 1989–1997.

5.2 CHANGES IN PROJECT RELATIONSHIPS

When analysing the archives of the focal organisation, I found only one document including an explicit analysis of what had been considered as key changes in fault diagnosis research and development. Figure 10 shows a part of this document. It illustrates how the focal organisation viewed the past and possible future evolution of its fault diagnosis relationships and skills in Spring 1993. The former are indicated by the names of the projects (e.g. “CORTEX”) and the interacting parties (e.g. “CERN”), the latter include different technologies (“distributed expert systems”), knowledge engineering and computer system techniques (“rule-based techniques, control system techniques”), applications (“energy management”) and indications of a problem solving process to be followed ("model-based approach"). Interestingly, the functional dimension that later was considered to be crucial for the fault diagnosis competence, is not mentioned at all.

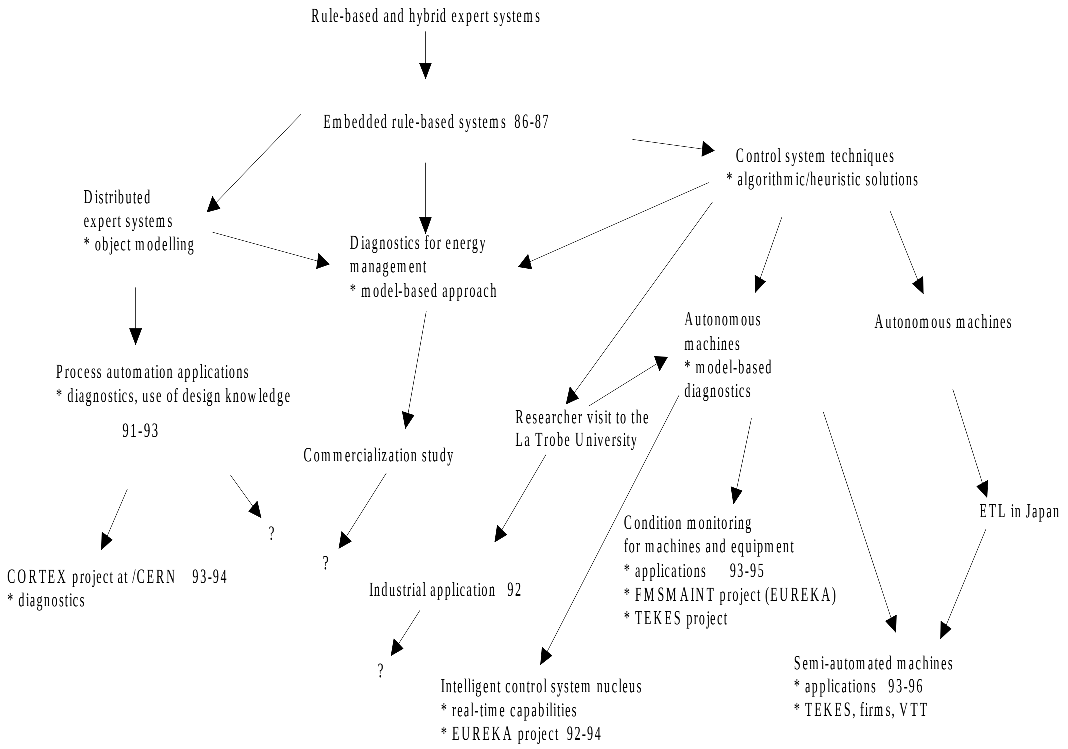

Figure 10. Illustration of the evolution of fault diagnosis R&D.

I will first analyse the change of fault diagnosis project relationships from 1989 to 1997, using the concepts of the substance and dynamic layers of the competence evolution framework. The initial period of 1986–1988 discussed briefly in [Seppänen et al. 1998a] is omitted, since it did not notably involve any external relationships concerning fault diagnosis. In the late eighties, the national funding body Tekes had established a research program on knowledge engineering techniques, however, VTT did not participate in the fault diagnosis projects carried out in that program. Most of the knowledge engineering related activities of the focal organisation at the time were joint blue research projects addressing intelligent software development tools. The initial red fault diagnosis projects could not be specialised from the blue project networks, but were started separately (cf. Figure 7).

1989–1991: Balancing of particularity and mutuality in blue projects

The focal researchers established bonds with knowledge engineers of process and machine automation companies, in order to acquire application knowledge and deploy knowledge engineering skills together with the customers' experts. Creation and exploitation of fault diagnosis skills was not a specific focus of the projects. The VTT personnel were neither closely associated with the customers' automation engineers nor with the end users, machine and process operators and maintenance people. Instead, application and system usage related knowledge was "acquired" from the end users together with the customers' knowledge engineers. Special knowledge engineering techniques and expert system technologies were then applied to make this knowledge explicit via computer systems.

As far as VTT was concerned, the relationships featured a high degree of mutuality and particularity at the same time (cf. cell 2 in Table 5). The focal organisation aimed at creating expert-to-expert bonds with industrial knowledge engineers, some of whom were former university and VTT researchers. The customers' knowledge engineers were often affiliated with research departments. Similar organisational structures helped to increase mutuality, which was necessary to enable the co-operation of people who were specialists in the same field of engineering. Some very close personal bonds were established between the collaborating experts. High mutuality was necessary to approach jointly the customers' application engineers and end users, i.e. the internal organisational customer structures of VTT's customers. Since the customers' knowledge engineers were a minority in their own organisations, collaboration with VTT may have increased their visibility and prestige, also on the organisational level.

Relationships involved during the period from 1989 to 1991:

- Strong bonds between industrial experts and VTT researchers,
- Many activity links and patterns, mostly in joint blue research projects,
- High particularity of resource collections and ties (e.g., expert systems),
- High capability due to focus on a few knowledge engineering techniques,
- High mutuality of activities due to bonds and webs between experts, and
- No considerable inconsistencies in mutuality, due to similar backgrounds.

The relationships can be characterised as research project networks. Blue projects were created from scratch rather than by exploiting (cf. Figure 7) green projects. Since the collaboration was based on interest in knowledge engineering techniques and expert system technologies, a novel approach to building industrial software applications, the purpose of both VTT researchers and the customers' knowledge engineers was to cause technical discontinuity. This ensured high mutuality of activities and particularity of resources involved in the collaboration. There were several successive projects with some firms and the parties co-ordinated mutually their interaction towards the customers' application engineers.

1992–1994: From blue actor bonds to red activity links

From 1992 to 1994 relationships stayed in cell 2 in terms of Table 5. Mutuality of activities and particularity of resources were rather high. Mutuality and particularity decreased, however, since many of the former personal bonds with industrial knowledge engineers came to an end, and organisational activity links involving customers' automation and application engineers emerged instead.

This was not only due to the economic recession hitting Finland in the turn of the nineties, but also a consequence of a disappointment over the lack of practical applications of early expert systems [Husso 1993]. The focal organisation needed to build completely new relationships in terms of customer representatives and collaboration schemes. Customers' customers caused a pressure for automation firms to address fault diagnosis. VTT considered automation as one of the key embedded systems research and development target, and was keen on offering services. The role of public funding bodies was rather distant, yet important. They financed a few blue research projects, but were not deeply involved in co-ordinating them. Blue networks became thus actually pseudo-red dyadic relationships.

Although most of the customers were still automation firms, the people involved in the projects were professionals in other fields, not in knowledge engineering. Even more significantly, they were product developers and not industrial researchers. Many of them had no personal ties to the focal researchers. VTT hired a few automation engineers to participate in the projects. This helped to deepen relationships with the firms that considered the basic knowledge of automation systems and applications a prerequisite for contractual co-operation. The research group responsible for fault diagnosis R&D activities reached a dominating state in some relationships, through a series of projects that led to solutions used in new products.

The relationships of the period from 1992 to 1994 involved:

- Red product development projects, as opposed to researcher bonds,
- Inter-organisational activity links to utilise application knowledge,
- Rather high particularity of resource ties (e.g., fault diagnosis solutions),
- Rather high capability in order to develop commercial products,
- Decreased mutuality of activities (mostly firm-level interaction), and
- A few inconsistencies in mutuality (automation engineers hired to help).

The relationships involved team-based and organisational collaboration between VTT and automation firms. Some relationships continued several times (cf. Figure 7), because the customers wished to improve the competitiveness of their products by using fault diagnosis functions VTT had developed in fully contractual projects. However, the mutuality of activities decreased since organisational activity links were emphasised instead of personal bonds between VTT and the customers’ researchers.

1995–1997: Encountering inconsistency in a new red network

By 1994, VTT can be said to have started to move away from cell 2 of Table 5. The organisation had generic capabilities for solving fault diagnosis problems. The first considerable green projects were carried out. The profile of the focal organisation was different comparing to the profile of some other parties that had become interested in the same area. The roles and organisational structures of VTT and its customers fit well together. Some universities, other VTT institutes and researchers of the focal organisation also participated in projects. However, universities lacked the managerial practices of VTT, and the other VTT people were not experts in fault diagnosis. End users, machine operators and maintenance personnel, were involved in some projects. The customers were on the whole satisfied with the projects, with the exception of one firm (cf. [Seppänen et al. 1998a]), in whose opinion VTT did not understand the customer's needs.

A red customer consortium project on GSM network diagnosis was initiated, since VTT wished to exploit the fault diagnosis competence in a very rapidly growing application domain. Customers' customers, i.e. telecommunication operators, were not involved. A situation similar to most of the earlier fault diagnosis projects. A funding body was involved by financing the industrial partners of the project, as a part of a national research program. The mutuality of activities and particularity of resources of VTT were quite low and inconsistency high, due to the lack of application knowledge that appeared to be very difficult to acquire.

The relationships established from 1995 to 1997 involved:

- VTT, research partners, automation and telecommunication firms and a funding body, interacting as webs of organisational actors,
- New inter-organisational activity patters and extended existing patterns, built in part by utilising new activity structures within VTT,
- Rather high particularity of resource constellations in automation related projects, low particularity of the resource collection of the GSM network diagnosis project (a distinct prototype was built for demonstration),
- High capability in automation related projects due to portfolio-specific competence, low capability in the GSM network diagnosis project (no commercial use of the developed solutions),
- High mutuality of activities in automation related projects, low mutuality in the GSM network diagnosis project (a consortium project), and
- Inconsistencies in mutuality in the GSM network diagnosis project (mastering the new application domain was slow and demanding).

The differences between the relationships established with automation and telecommunication firms are rather obvious. The former included several red projects, occasionally expanded (cf. Figure 7) to red networks involving many parties. The latter aimed at capturing new customers, by exploiting (cf. Figure 7) internal green activities to a contractual red network.

5.3 EVOLUTION OF THE FAULT DIAGNOSIS COMPETENCE

The fault diagnosis competence evolved considerably within the project relationships. This will be demonstrated in the following.

1989–1991: Absorbing knowledge engineering skills

Fault diagnosis was regarded in the focal organisation as one possible problem area for deploying generic knowledge engineering techniques. A lot of the related knowledge was explicitly coded and generic. Relationships established with knowledge engineers of process and machine automation companies resulted in the combination of the new knowledge with the formerly tacit application knowledge ("expertise"). Knowledge of the implementation technologies of automation systems was not of significant importance, while the so called expert system technologies, including special expensive system design tools, were emphasised. Knowledge engineering techniques and expert system technologies were believed to become the dominant computing paradigm of the nineties. At the time they were in the substitution phase, and needed to be "embedded" into more traditional computer technologies (cf. Appendix 1).

VTT aimed at absorbing skills in knowledge engineering techniques and expert system technologies to build fault diagnosis features into research prototypes of new automation systems. Tacit application knowledge was scanned from the customers' automation engineers and end users together with the customers' knowledge engineers. Its explicit representation and utilisation required special tools, since the computer technologies on which automation systems were based changed only incrementally. In 1990, the annual report of the focal organisation described the situation as follows:

> *"Our research in embedded knowledge based systems expanded. Both knowledge based control and diagnostics systems were developed for companies in machine automation and process industries."*

The fault diagnosis competence evolved in 1989–1991 as follows:

- KE techniques were absorbed to prototypes of automation systems,
- Tacit application knowledge was scanned from customers' experts,
- Expert system technologies were used to cause technical discontinuity,
- Incrementally changing computer system technologies were used, and
- Fault diagnosis functions were developed for the system prototypes.

The competence can be characterised as project-team based, mostly tacit and context-specific skills to solve fault diagnosis problems by using generic knowledge engineering techniques and expert system technologies. Tacit application knowledge was acquired, made explicit and combined with technical knowledge. They were used to investigate solutions to fault diagnosis problems. VTT researchers could also utilise their knowledge on the incrementally developing computer system technologies.

1992–1994: Creating portfolio-specific skills through problem solving

The competence of VTT moved towards a much wider engineering area compared to a prior claim to knowledge engineering techniques and expert system technologies. The former were still used, but now in a supporting rather than a central role. VTT wished to extend it capabilities in building the so called embedded expert systems. The customers' knowledge engineering skills had not increased, due to the rapid fall of industrial interest in the techniques. At this point VTT had already made explicit, through problem solving in successive projects, some knowledge of automation applications, fault diagnosis functions and computer system technologies, which can be considered as a portfolio-specific resource collection. This may have created a barrier to competitors (cf. [Seppänen et al. 1998a]), e.g. to the customers themselves, who possessed mostly special application knowledge, and to universities, whose knowledge involved mostly generic engineering and scientific techniques.

Regarding knowledge engineering techniques, the increasing number of distinct projects meant that VTT was still absorbing application knowledge, although explicit information, e.g. design documents, were becoming more important than tacit human knowledge. The expert system tools used in establishing and carrying out some of the earlier projects became obsolete, they did not cause any technological discontinuity [Husso 1993]. They were instead substituted by – or lost competition to – the computer system technologies routinely utilised by industrial automation system designers. These technologies were changing, but only incrementally.

The fault diagnosis knowledge had clearly been diffused to the focal research group and not only to individual project teams. The organisational knowledge started to further develop in new projects established with automation firms. The resulting solutions can be viewed as fault diagnosis systems, as opposed to fault diagnosis features of automation systems. The solutions involved commercial products instead of research prototypes.

The competence evolved in 1992–1994 as follows:

- Knowledge engineering techniques lost their central position,
- Explicit application knowledge was absorbed for fault diagnosis,
- Computer system technologies replaced expert system technologies,
- These technologies remained in their incrementally changing state, and
- Fault diagnosis systems were developed as parts of commercial products.

The competence can be characterised as portfolio-specific combination of skills of an organisational team to create and use resource collections consisting of computer system technologies and knowledge engineering techniques. Concrete fault diagnosis systems were built with the help of customers' automation engineers. Explicit resource ties and activity links based on joint product development thus replaced most of the former actor bonds with industrial knowledge engineers built in joint research projects.

1995–1997: Building and using of a core fault diagnosis platform

The portfolio-specific fault diagnosis competence helped VTT in extending its relationships. The focal research group indicated that it offered a comprehensive process to solve industrial fault diagnosis problems. A core fault diagnosis platform emerged gradually as a resource collection of VTT, around a set of system design tasks including modelling of the application, design of the fault diagnosis functions, testing of the functionality by actual data, and implementing the system. The platform was documented in [Kurki 1995], by a group manager involved in many fault diagnosis projects.

Problem solving based on the core platform became important, in addition to the absorption of application knowledge used to built fault diagnosis systems. This was described in a project proposal in 1992 as follows:

> *"A project has been carried out together with [...], funded by Työsuojelurahasto, on fault diagnosis of a mechatronic [...] machine. From the results of this project it has been seen that a system based on generic notations can be developed, to be applied as such or at least in part in different types of mechatronic machines and equipment."*

The core fault diagnosis competence was made explicit through a set of functions, in order to separate the platform from specific automation applications and their implementation technologies. A pool of generic knowledge engineering techniques was used to support these functions. It could be updated on the basis of the incrementally changing life cycle of the techniques. For example, fuzzy logic, case-based reasoning and neural network techniques were learnt by absorbing, to update the pool. Even more importantly, generic computer system modelling techniques helped to link fault diagnosis functions to applications and implementation technologies. Customers' experts were used to aid in application modelling and understanding implementation constraints. A set of system design tasks created a problem solving frame that combined the different areas of skills. The tasks are summarised in [Kurki 1995] (page 207, Figure 28): modelling, configuration and adjustment of fault detection, representation of the diagnostic knowledge, and learning, evaluation and testing of the system.

Even though there were other people from VTT and universities participating in some projects, the functional fault diagnosis problem solving skills did not diffuse outside, but remained within the focal research group. The focal researchers and managers had learned through practice, the meaning of solving fault diagnosis problems for industrial products. The generic fault diagnosis functions that were a heart of the platform, represented an abstraction of these skills and were difficult to diffuse to outsiders. Even though Figure 11, taken from [Kurki 1995], is rather technical, I decided to include it in the dissertation. It illustrates the model-based approach to fault diagnosis that emerged as a basis for the core fault diagnosis platform, using the concepts of the subject matter. The competence evolution framework, as presented above, makes it possible to model aspects of competence in similar details, an issue I consider to be important when analysing changes taking place in small organisational and project teams as key actors.

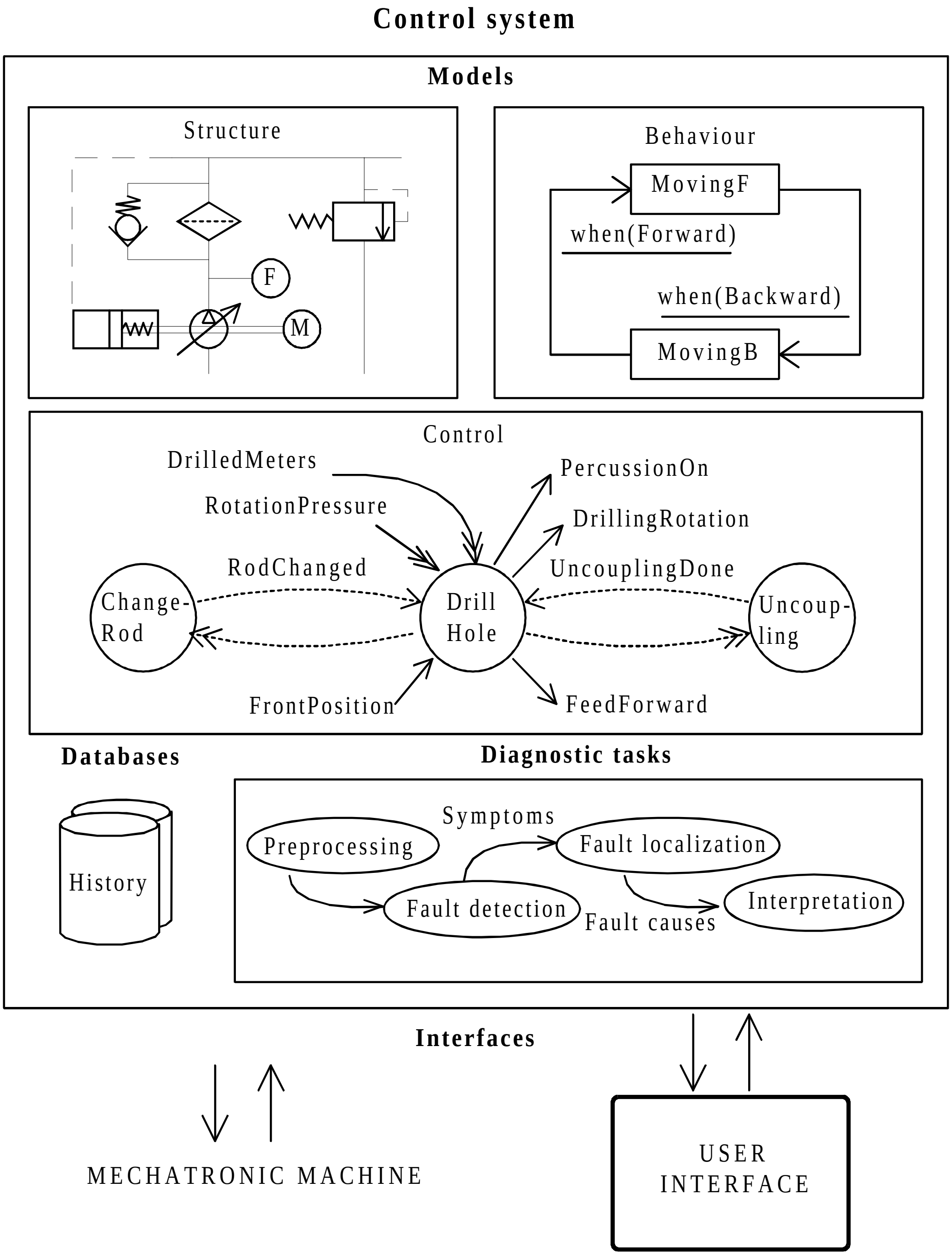

Figure 11. The model-based approach to fault diagnosis [Kurki 1995].

The “models” of the system to be diagnosed ("mechatronic machine") include so called structural, behavioural and control models. The former are based on the design methods and concepts used in the application domain, the two latter ones on embedded computer systems design methods and concepts. The "diagnostic tasks", as they are called in the figure, form the functional core of the approach. The "user interface" illustrates the fact that the developed fault diagnosis systems became parts of larger products used by machine operators and other end-users, rather than technical experts. The label “history” in “databases" shows that instead of mere expert knowledge, data acquired continuously from the operational machine is used to diagnose faults, by associating it with the models of the expected operation.

The first GSM telecommunication network diagnosis project was based on the idea of specialising the developed approach and core platform for a new application domain. This idea was sold to a few interrelated business units of a single customer company. One of the customer representatives described the initiating of the project in an interview as follows:

> *"Alarm management and diagnosis had been identified as an area where new ideas, techniques and perhaps skills that we do not possessed were needed. This was also an area that crossed the boundaries of the product lines [of the company]. VTT, especially [the focal group manager] succeeded in presenting convincingly new ideas and the existing competence of VTT."*

However, the application knowledge appeared to be highly tacit, difficult to absorb from documents and acquire by scanning from experts. Problems arose even concerning the notations through which it could perhaps be made explicit, not to mention the content itself. The new knowledge engineering technique, case-based reasoning (CBR), was absorbed from the literature. CBR was supported by special development tools, which did not fit with the implementation technologies and tools used to design the network equipment. The situation was thus much similar to the early phase of the fault diagnosis research, with the exception of the customers not having any knowledge engineers with whom to associate. It seems that too many things were changing at the same time: not only the application, but also the pool of generic knowledge engineering techniques and to some extent the computer system implementation technologies were changed, from VTT's viewpoint. It was difficult to model the application that consisted of many equipment, although VTT had gained experience from the model-based approach to solving fault diagnosis problems, and was following it in other projects being carried out for automation firms.

The types of system models produced for automation applications (cf. [Kurki 1995]) were not feasible in the new application domain. Therefore new modelling techniques needed to be invented. In any case, the customers were happy with the new techniques, since they provided for a comprehensive but yet compact view to GSM networks. However, the customers were questioning the importance of the CBR technique. The focal researchers were not familiar with the technique, and thus had to spend a lot of time for absorbing explicit knowledge related to it. Only a simple prototype for demonstration purposes could be built based on the technique and the acquired application knowledge. The core fault diagnosis platform could not be utilised in the new domain, although it was being heavily used in several projects being carried out for automation firms.

By conquering the new application domain, VTT had been aiming at an incremental change in fault diagnosis competence. VTT fell back into an early substitution state – fighting against dominant but partial GSM network equipment monitoring solutions by using the model-based fault diagnosis approach and the CBR technique. The intended organisational competence building in the red consortium project failed and was replaced by extensive knowledge absorption, mainly carried out on an individual basis.

In an interview, one of the VTT managers compared the automation and GSM telecommunication network diagnosis projects to each other:

> *"Speaking of [the GSM network diagnosis project], something was tried to be done for many several business units, concerning products that had existed already for a long time. It is a very hard job. The topic itself is still valid, but it is a different question what can be done for it. It is easier to do something to new single products that have no similar solutions yet, and are managed by one company."*

The fault diagnosis competence evolved in 1995–1997 as follows:

- The CBR knowledge engineering technique was taken in use,
- Special system design tool technologies were taken in use,
- Explicit and implicit GSM network application knowledge was acquired,
- New application modelling techniques needed to be invented,
- A prototype was built for demonstrating GSM network diagnosis, and
- Platform-based fault diagnosis systems were built for automation firms, using the dominant model-based approach.

The core fault diagnosis competence was utilised in several projects carried out for automation firms, to a great extent along the earlier lines. Knowledge on the new GSM network application domain, its modelling techniques and the CBR technique were created on an individual basis. The existing resource collection of VTT, the core platform, could not be reused. Only a prototype was built to demonstrate elementary fault diagnosis features. It could not compete against the dominant but partial functional solutions developed by the customers themselves. The activity links and patterns established with the customer consortium remained very weak.

5.4 PRELIMINARY EMPIRICAL FINDINGS

In this chapter I have analysed the evolution of the fault diagnosis competence of the focal organisation in project-based research and development relationships. The problem is reciprocal, since relationships will not, in practice, evolve without changes of competence. This can be seen, for example, in the evolution of bonds between VTT researchers and customers' knowledge engineers to bonds between automation and process engineers. The main reason for this development was the change of the use of knowledge engineering skills in fault diagnosis. A preliminary version of the layered competence evolution framework presented in Chapter 4 was applied to study the evolution of the competence in relationships. The framework was used both to model and analyse the fault diagnosis case. A few additional models can be found in [Seppänen et al. 1998a]. They are not included here, since their purpose was rather to illustrate the competence evolution framework than to analyse the case.

During the analysis of the fault diagnosis case, a number of empirical, theoretical and managerial issues concerning the case, as well as the framework and its further development, arose. These will now be discussed, in addition to the problem concerning contexts, related to the process of studying industrial networks.

The analysis of the fault diagnosis case was largely based on a summary put together by myself and commented by a few key informants (cf. Appendix 1). In [Seppänen et al. 1998a] the summary was augmented with texts illustrating aspects related to individual actors, activities and resources. This kind of augmentation was workable, since it resembles a real life situation in the sense that the organisation's understanding of the past forms a backbone for distinct episodes. Although the augmentation was in part based on interviews of the involved people, I mainly used documentary data to construct the case summary. This was necessary to address individual projects and the resources created and utilised in them.

The case data was organised into a few three year phases. The first and the last phase were strongly affected by generic knowledge engineering techniques. In between, VTT focused on creating and applying fault diagnosis skills in connection with a growing portfolio of projects involving industrial automation systems. One of the reasons facilitating the expanding of relationships in this phase was the increasing automation of processes and machines. Moreover, the needs of some end users to better utilise their expensive machine investments affected the developments. This was described by two VTT managers in interviews as follows:

> *"Yes, the first concrete diagnosis efforts were certainly the ones done for [...]. I talked to [...], the R&D manager of [the customer], which is still an organisation that must be pushed by their customers. They will not develop any automation voluntarily. They are hardware twisters, they still talk about bird-houses: automation engineers come and spoil their good solid hardware by installing some bird-houses [embedded computers] to their products. The number of advanced customers that would request more automated features is, however, not high."*

> *"The customers may have pushed them. Themselves they of course did not have any expertise on knowledge engineering or diagnosis".*

A fault diagnosis platform, an explicitly codified piece of the focal organisation's competence, resulted from carrying out several successive projects. The strategic plan of the focal organisation written in 1992 includes a brief section on the evolution, present status and future outlook of knowledge-based fault diagnosis research and development activities:

> *Strategic plan of VTT Computer Technology Laboratory 1992, H. Hakalahti, 5.5.1992. "Research on embedded knowledge-based systems was started at the laboratory in 1986. The first practical result was achieved in the machine control application in 1989. This line of research was continued in a large machine automation related project that will end in 1993. Development of diagnostic systems for machine automation started in 1990."*

The core of the platform consists of certain diagnosis functions. Specific applications, implementation technologies and knowledge engineering techniques could be associated with the functions by certain system design tasks forming a systematic problem solving process. One of the above mentioned managers characterised the competitive advantages of VTT in automation related fault diagnosis projects as follows:

> *"We had superior knowledge of expert systems and other techniques that could be used to solve this kind of problems. Although we did not have the application knowledge, this [automation] was an area were AI [artificial intelligence] was not day-dreaming. Moreover, there were practical technologies available, supercomputers and hundreds of man-years of work were not needed. We had the generic knowledge of methods and tools that could be used when diagnosis problems emerged. Application problems were concrete and simple enough. It was possible to proceed through small steps."*

Application knowledge was utilised together with customers' experts based on models created from design documents, using methods and tools familiar to these experts. This was a remarkable improvement:

> *Final project report, 1993: "The existing diagnosis systems can be roughly divided into first and second generation systems. First generation systems are usually rule-based [expert system] implementations, which contain only shallow knowledge of the machine and its operation. The design process is laborious and systems are difficult to maintain, a minor change in the structure of the machine may require a total redesign of the diagnostic software."*

The rapid emergence of telecommunication applications in the late nineties seemed to offer an excellent opportunity for creating new relationships by reusing the core fault diagnosis competence. The difficulties that were encountered in this domain were mostly due to the lack of application-specific knowledge scattered over the customer's business units. These units had existing partial solutions competing against the proposed model-based approach. Since problems arose with modelling of the application and associating the CBR technique with the models, it is no wonder that competence evolution seemed to stall in this domain.

Focus on generic knowledge engineering techniques was useful especially in the first phase, but corrupted rather rapidly due to the decreasing industrial interest. At the beginning, the knowledge engineering skills depended on special tool technologies and close interaction with customers' fellow experts. A group manager of the focal organisation evaluated the situation of one of the former key customers as follows in March 1992:

> *"Organisation [of the customer company] has changed at the beginning of 1992 so that R&D is not a profit centre. This means that corporate funding will be reduced, which will affect considerably the previously loose research and development policy. Possibility to develop products together with end users. AI programs developed for several years will become useless, if PC platforms are taken in use."*

Focusing on fault diagnosis functions and employing the application engineers helped VTT to build a dominant fault diagnosis system design for automation applications. The role of customers' application experts matched with the role of the VTT researchers in mutual actor bonds. The model-based approach that the customers could understand was of great help in creating strong activity links. However, one customer firm did not consider the approach as useful for its needs. The firm would have been interested in trying new knowledge engineering techniques. [Seppänen et al. 1998a] also indicates that some firms were not interested in fault diagnosis because of the technological immaturity of their products. There were some harsh comments even from firms that later on became key customers, indicating that they suspected the skills of VTT and the need for fault diagnosis:

> *Jukka Karjalainen's R&D diary notes, 12.5.1993. Visit at a potential customer firm: "There is no use to start co-operation, because communication and learning of things takes so much time. They have themselves top-level expertise and strong self-respect, how external parties could know things better. A state organisation does not understand anything of what it is to carry out real business in the time of a recession. Diagnostics: a new generation of diagnostics is under consideration, but he [the customer's R&D manager] is somewhat pessimistic, because condition monitoring has no importance."*

The fault diagnosis platform, a product concept, resulted from the problem solving process being carried out in several projects. Interestingly, Tekes, having funded most of the focal organisation's blue research projects, financed hardly any fault diagnosis research. Instead, the Finnish Work Environmental Fund was used – diagnosis was motivated by safety-related reasons. The projects were not much controlled by the funding body, the competence could be developed quite freely in several successive projects.

New implementation technologies were used in almost all projects. Technology, as a means of implementing fault diagnosis functions, was thus not unimportant, but did not play any central role either, after the fall of special expert system technologies. This is certainly one of the most interesting empirical findings in the case. Fault diagnosis systems were after all implemented by using incrementally changing rather than revolutionary technologies, and this was useful for close resource ties. New knowledge engineering techniques were absorbed for their possible utilisation.

On the whole, one of the main reasons for change has been the evolving of project relationships from joint blue knowledge engineering research projects to red fault diagnosis system development projects carried out especially for automation companies. Matching of the roles of VTT as a contract research organisation with the roles of other actors in actor bonds has also been important. Weak actor bonds and activity links, as well as resource ties based on entirely new techniques and technologies have proven to be problematic. In this regard it is interesting to point out that the ideal technology push pipeline from green and blue to red projects has not been obvious. For example, there were hardly any self-funded green projects until in the late nineties, and the fault diagnosis platform emerged largely as a spin-off of blue (pseudo-red) and red projects.

5.5 INITIAL THEORETICAL CONTRIBUTIONS

The main theoretical contribution of this part of the thesis is the initial competence evolution framework. Although longitudinal analyses of industrial relationships are not any more uncommon, they often do not address competence in depth. Even Rosenbröijer [1998] investigating the problem thoroughly remains at a rather abstract level with regard to competence. Moreover, he focuses on certain relationships of the focal organisation rather than on its competence. The reversed approach I employed by analysing the relationships of the focal organisation concerning specific competence, is oriented towards the basic business strategy of contract research organisations. The goal of VTT, for example, is to create, maintain and extend relationships for making use of its competence. The number of such competence is nearly invariably smaller than the number of relationships, because of the need to transfer new technical expertise to industry as a whole and not only to few specific firms.

Although the substance layer of the framework is based on the well-known ARA model, it contributes to the model by including a conceptualisation of the resource dimension. The R&D resource typology I used to analyse the fault diagnosis case is based on an earlier work. All the same, I revised and extended the work and associated it with the concepts of R&D actors and relationships. In particular, the four-dimensional model of the content of competence related to the creation and use of human, technological and physical resources – applications, functions, techniques and technologies – proved to be useful. Even though this model is based on [Abell 1980], its use in the context of competence of contractual R&D is quite novel. Elfring and Baven [1996], for example, miss two of the four dimensions. The dimension of generic techniques not included in Abell’s original market model played a crucial role in the developments of the fault diagnosis case.

After all, the substance layer of the framework is still a yet another deployment of the ARA model. Although it is interesting as such at least for the needs of better understanding contract research and development, the main contribution of the framework is the dynamic layer. Using [Ford et al. 1986] as a starting point, I identified a number of possible changes in relationships and four functions to manage the changes. Earlier research has addressed changes of relationships and networks, as well as investigated their reasons and results. However, I have provided a relationship-based view to competence evolution in the context of contract R&D that is still very rare, as pointed out by Reger and Schmoch [1996].

Although the typologies of the substance layer of the framework are hierarchical, there is a need to develop a more elaborated structure in order to focus on truly critical elements of relationships and competence. Too broad a categorisation of a large number of concepts involved in real-life relationships makes the analysis complicated. The kind of a problem solving process that was developed in the fault diagnosis case can be used to structure the substance of competence and relationships. I will propose a process frame to be included in the framework in Chapter 6.

5.6 FIRST MANAGERIAL IMPLICATIONS

One of the basic managerial problems of a contract research organisation is to know how to employ its knowledge and skills in a “correct” way in order to open up and manage its position in a complex portfolio of projects. Competence should be actively developed and exploited, and at the same time the dangers of *core rigidities* should be avoided [Leonard-Barton 1992]. This depends on the particularity of resources and mutuality of activities: "appropriately" strong, neither too loose nor too tight, resource ties and activity links are required.

The fault diagnosis case indicates that the development and deployment of the focal organisation’s competence was a highly concurrent process. There were no clearly distinct development and exploitation phases, as opposed to the view of e.g. Sanchez and Thomas [1996]. On the contrary, it seems that although the ideal project portfolio of the focal organisation separates competence building *for* red relationships in green and blue projects from competence leveraging *in* red projects, such important developments as the emergence of the fault diagnosis problem solving process and the core platform resulted from simultaneous competence exploitation and development activities. The activities were mainly employed in red projects.

Concurrent management of present and future competence would thus be needed. The managers of a contract research organisation should be able to navigate in the space of current and new competence, in terms of evolving relationships. An effort to illustrate this is shown in Figure 12. The fourth “mass-customised R&D” cell, in which the organisation takes an advantage of its existing competence, while being simultaneously active in the development of new competence, is a good target state. The path to that state depends on how the organisation and its customers view and develop their relationships and how their competence blend in and evolve together.

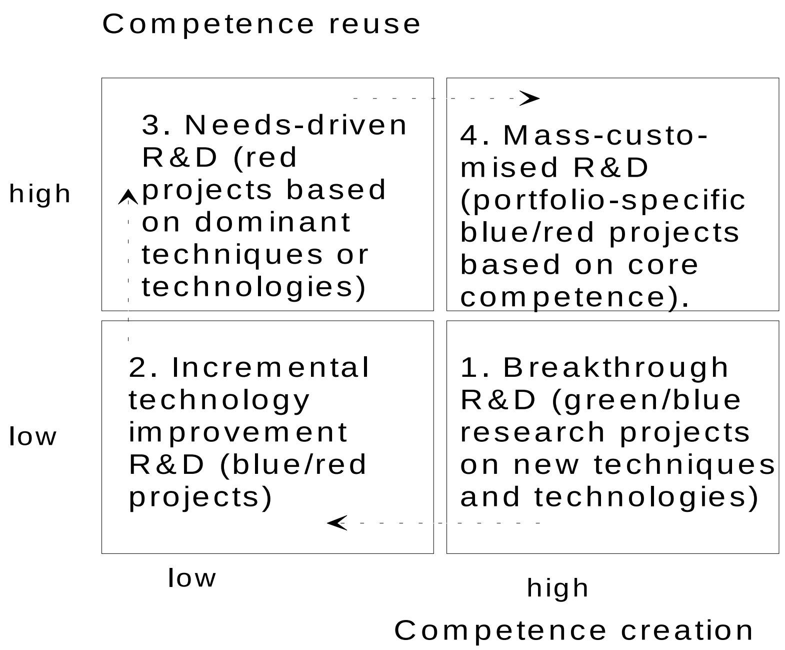

Figure 12. Competence management in contract research and development.

The evolution of the four dimensions of the content of competence, applications, functions, techniques and technologies, is crucial for the change, as the fault diagnosis case clearly illustrates. The use of the four functions by which desired changes of competence can be co-ordinated should be analysed in more detail. They can be expected to result in an increased “maturity” of competence, depending on changes of the life cycles of the four dimensions (cf. [Elfring and Baven 1996]):

- in an "early" phase, competence is based on knowledge of radically new (cell 1, “breakthrough R&D”) or incrementally improved technologies (cell 2), or on the understanding of techniques not yet applied to solve certain problems or not yet implemented as some commercial technologies, whereas
- in a more “mature” phase competence has evolved to core platforms (cell 4), resulting from the understanding of the problem solving process, i.e. reuse of core functions, use of dominant techniques and implementation technologies, and better understanding of certain customer applications (cell 3, “needs-driven R&D”).

This would mean that the increase in the maturity of competence builds on the top of technologies and generic techniques, nevertheless focusing on functional skills and processes needed to solve certain problems for the current key customers, by means of actively investigating possible new applications for those skills. In this process, technologies and generic techniques are used for ensuring technical methods flexible enough for solving problems. If techniques or technologies are new, they and the supplier's functional skills must be accepted by customers (cf. automation firms in the mid-nineties).

If the customer has already developed functional solutions for the problem (cf. the GSM network diagnosis project), the situation is more difficult. If the solutions are based on well-established techniques and technologies in which the customer is an expert, the challenge the supplier faces is indeed notable. The supplier can adapt to the situation, as in the automation related fault diagnosis projects in the nineties, or try to cause a technical discontinuity, as happened in the early automation related and in the first GSM network diagnosis project. In terms of [Gallon et al. 1995] techniques and technologies could be called *primary capabilities*, whereas *critical capabilities* would be the ones used to develop and exploit core competence for solving problems on which the supplier wishes to focus. The knowledge of the customers’ applications and end-user needs would be either critical or primary, depending on the importance of a specific customer account and the relationships between the customer and its own customers.

In order to manage change paths in the competence evolution space in practice, the supplier organisation needs to carry out its strategic and operational planning based on a vision of which new elements could be incorporated into the content of its competence within the emerging portfolio of relationships. Table 7 illustrates such planning, by showing the plans and results of recent fault diagnosis competence evolution.

Table 7. Planning and realisation of competence evolution.

Fault diagnosis competence	**New applications**	**New functions**	**New techniques**	**New technologies**
1998–1999	Electronics assembly lines: *tried in a blue project* Network operating: *a red project is ongoing*	Data mining, optimisation: *are being integrated with fault diagnosis functions in blue projects*	Neural networks, SPC, genetic algorithms: *not yet been utilised*	Assembly automation, neural networks, data bases: *only data bases are being utilised*
2000–2001	Intelligent electronic instruments: *a red project is ongoing*	Explanation, visualisation: *several blue projects are ongoing*	Usability engineering, multimedia engineering: *green, blue projects are ongoing*	Multimedia, visual user interfaces, virtual reality, WWW: *used in several projects*

Despite the problems, VTT managed to continue industrial co-operation in GSM network diagnosis, but only with one of the network equipment manufacturers. VTT is now heavily utilising its core fault diagnosis competence based on the model-based approach – for a new generation of equipment in which alarm handling and monitoring can be designed much anew. Furthermore, VTT has also carried out the first red fault diagnosis project for a network operator – the customer's customer.

In 1998 a market study was carried out on intelligent instruments, where fault diagnosis functions could be included. The study was performed to continue and extend the work carried out in a green project for a case provided by a small company. About fifty companies were contacted, ten of which were interested in the topic. However, none of them could finally take part in the planned blue research project. Nevertheless, opening of this new application domain of model-based fault diagnosis continues as a rather large red project contracted by the above mentioned company from VTT.

In addition to the planning of competence evolution, it should be known how to evaluate the “maturity” of certain competence. Maturity assessments have become an industrial practice in some fields, e.g. software industry. They have also been analysed from the viewpoint of research and development (see, for example, [Miller 1995]). However, the existing models focus on internal activities rather than on external relationships. For this reason they do not take the changing context of the organisation into account. Such approaches as [Lewis and Gregory 1996] could perhaps be used as a better starting point for developing a context-sensitive competence maturity assessment framework for research and development.

5.7 PROBLEM OF CONTEXTUAL ANALYSIS

The importance of contexts for understanding firms and business relationships is generally accepted (see, for example, [Lundgren 1992, Håkansson and Snehota 1995, Halinen and Törnroos 1995, Alajoutsijärvi and Eriksson 1998]). To name a few, studies that have addressed contexts as part of industrial relationships include [Anderson and Narus 1994, Halinen 1994, Alajoutsijärvi 1996, Rosenbröijer 1998]. In this chapter I have used the ARA model to make explicit a specific context of contractual research and development, while not analysing contexts more thoroughly. However, in order to prepare for addressing contexts in greater detail in the third part of the dissertation by using the revised competence evolution framework, I will include a brief discussion on contexts below.

Contexts are central in process-based approaches to industrial management [Pettigrew 1985, 1987, 1989 and 1992, Pettigrew et al. 1989 and 1992]. In these approaches, contexts usually describe a broader area of connectedness or embeddedness, compared to business networks surrounding the focal firm. Pettigrew [1987] considers contexts to be composed of economic, social and political forces and developments (the external or *outer context*) and a company's resources, capabilities, culture and politics (the internal or *inner context*). However, the authors do not provide any extensive discussion especially on what the inner context is, which has been pointed out in [Alajoutsijärvi and Eriksson 1998].

Different aspects of the inner context can be found in the existing relationship models. For example, notions of situational features of individuals, the social system and the atmosphere of relationship models can certainly be seen as elements of the inner context. Such relationship models are provided e.g. in [Möller and Wilson 1995a, Halinen 1994]. From my point of view, one of the basic problems with these models is that they try to make an a priori definition of what the relevant inner context is. Furthermore, their definition of the context is rather narrow. The complexity of the context is often used as an excuse, which leaves little room for case-specific variations [Alajoutsijärvi and Eriksson 1998].

Concerning the in-depth analyses of outer contexts, several macro forces affecting certain industrial relationships are identified in [Alajoutsijärvi 1996]. Halinen [1994] proposes "general social, political and technological forces" as the outer context of professional services.

I will use the results of the fault diagnosis case study discussed in this chapter as a starting point for making inner and outer contexts of contract R&D explicit, and for analysing how the two are interrelated. I will return to this question in the next chapter, when introducing the revised version of the competence evolution framework. The so called logic of action will be added to the framework, in order to describe the goals, means and justifications of activities carried out by actors, be they individuals, groups, organisations or industries as a whole. In particular, different logic may be aligned or conflicted with each other, which contributes to the basic dynamics of especially the inner context.

PART III: COMPETENCE EVOLUTION IN FOCAL NETS

The original purpose of the third part of the dissertation was to apply the competence evolution framework presented in Part II again to analyse the code generation case, an "unsuccessful" attempt to create and deploy core competence. Soon after I had started to work for the code generation case, it became clear that the chicken-and-egg problem had hit me once again. I could not analyse the code generation case by using the initial framework. However, the case data had to be used to justify changes of the framework.

6 REVISED FRAMEWORK

The initial competence evolution framework seemed, on one hand, to lack concepts to make explicit and analyse controversies among the individuals and organisations involved in code generation related activities. On the other hand, I ran into difficulties with the detailed typologies of competence and relationship concepts that I had developed for the initial framework (cf. [Seppänen et al. 1998a]). The code generation case involved a very large variety of aspects that could not be easily managed by the typologies. Therefore, I needed both to extend and simplify the initial framework.

For the latter cause, I took in use a process-based approach based in part on [Tikkanen 1997]. The approach facilitated considerably the tying of the key competence and relationship concepts together in a manner that supported the analysis of changes rather than static structures built by using typologies. To extend the initial framework, I adopted the idea of logic of action. I studied some other approaches, such as the actor-oriented viewpoint to innovation networks [Miettinen 1998] that has been used to analyse R&D projects carried out at VTT. However, in my opinion, the logic of action suited very well the ARA concepts that are the foundation of the competence evolution framework. Moreover, the logic of action had already been successfully used in business studies (e.g. [Eriksson and Räsänen 1998]). For these reasons I chose to employ it in the revised framework. I introduced this idea in the annual IMP conference [Seppänen et al. 1999b], to gain some initial feedback from fellow researchers.

6.1 PROCESS-BASED VIEWPOINT TO CONTRACT R&D

The substance layer of the framework presented in Chapter 4 makes the governance structure of relationships explicit through the actors and relationships of focal nets. In addition to this *organisational* viewpoint, the *product*-based viewpoint is included through certain types of resources. Although activity types are associated with resource types, the *process*-based viewpoint is not emphasised. The ARA-based model makes the basic competence-related concepts of activities and resources explicit, but it seemed to me that processes would help to link them closer together.

Processes would also facilitate viewing of relationships through the roles played by different parties in carrying out activities on resources. For example, a steering group of a project typically controls the use of the financial and temporal resources of the project. The steering group also accepts the technical results produced by the project team. Networks of relationships related to processes can be seen as *process nets*, as in Tikkanen [1997], i.e. the networks revolve around the processes being carried out by the participants.

6.1.1 Business processes and networks

I will use Tikkanen [1997] as a basis in introducing processes at the substance layer of the competence evolution framework. His work is one of the most recent and comprehensive studies involving a process-based viewpoint to industrial relationships: “the main idea explored in this book is whether it would be fruitful to study business processes through the increasingly rich conceptual arsenal offered by the industrial networks research tradition”. Tikkanen proposes the ARA model as a means of making explicit and analysing business processes, as shown in Figure 13. The business process, consisting of activities and resources, is viewed as a "production system" for the focal net of actors and their relationships.

At VTT both the R&D and business processes revolve around projects. Tikkanen [1998] points out three typical features to separate project marketing from other kinds of business marketing: discontinuity of demands for projects (emphasising long-term supplier-buyer relationships), complexity of individual projects in terms of the number of actors, and the uniqueness of each project with regard to technical, financial and socio-political aspects. According to Tikkanen “future projects are actively anticipated and constructed primarily trough social exchange within the relevant network during the ‘sleeping’ phase of the focal supplier-buyer relationship”. After having worked in an R&D project organisation for over fifteen years, I would like to add that although project marketing involves active social exchange, it is also a serious business endeavour at least for the supplier. Therefore, many organisational arrangements are also utilised, such as key customer accounts and strategic alliances.

Tikkanen sees project business becoming more specialised than earlier, “the nature of contemporary project business is characterised by an increasing uniqueness of individual, highly differentiated project deliveries... Thus, the potential for replication might be diminishing [as opposed to project replication]”. I doubt this view. For example, in the electronics industry not only small suppliers, but also big buyers aim at establishing strategic partnerships to agree on certain foci areas of co-operation. As an example, in the eighties VTT was selling projects to many different industrial fields and customer companies. In the nineties it has created alliances concerning special competence areas, so that a far more larger portfolios of projects could have been planned together with key customers.

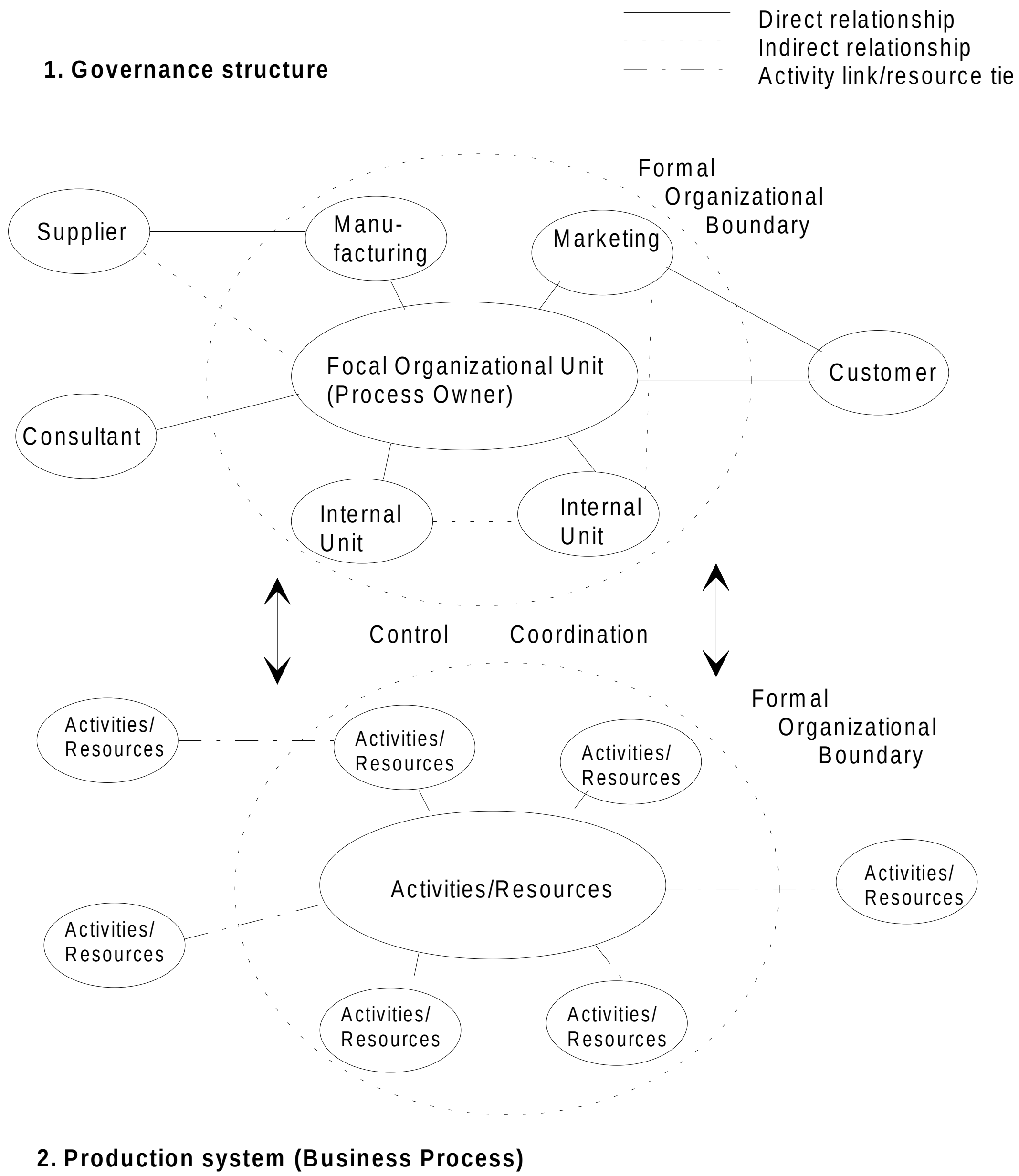

Figure 13. A hypothetical focal process net [Tikkanen 1997].

Tikkanen identifies four phases in the project-based business process of one of his case organisations: bidding or negotiation, planning, design and implementation. He studies actor involvement and activities performed on resources in each of these phases, and actually also in a fifth phase, transition. These phases represent the seller's point of view, i.e. the *project marketing cycle* (cf. [Tikkanen 1998], based on Holstius). From a slightly different perspective, Matikainen [1998] has studied the problem of governing different types of relationships efficiently. He approaches relationships from a contractual perspective and studies their efficiency. Three levels of efficient governance are identified for eight governance processes: screening, signalling, dependence balancing, bargaining, incentive planning, adaptation, monitoring and enforcement. They can be considered as particular types of co-ordination and control activities (cf. Figure 13) in interrelated business processes, viewed from the perspective of the efficiency of the relationships in which the processes are involved.

Tikkanen's [1997] claim that “individual project deliveries could be characterised as more or less unique” may be correct for his case company, but for VTT this is not true. The organisation aims at increasing its skills by utilising knowledge gained from the past projects to be used in subsequent projects. If all projects were seen as unique and no means could be established for learning something from successive projects, the organisation would soon become dominated by other network parties. However, as indicated in the fourth part of the dissertation, the matter may conceptually be quite complicated, even though it is managerially self-evident.

Tikkanen’s analysis of focal nets was based on interviews, in which “it proved to be impossible to track any more specific sequential activity chains [in one of the case organisations] than the rather general ones”. Furthermore, in “neither of the cases described and analysed in the focal net study was it possible to relate the resource dimension exactly to the specific activity structure of the focal process net”, concerning especially “the more abstract intangible resources, such as technical know-how and personnel capabilities.” In the code generation case discussed in the next chapter I also found it very difficult to acquire facts related to individual activities and their results. It was especially difficult for managers to remember factual information, possibly due to their wide sphere of responsibilities. In Tikkanen’s study, three of the five interviewed persons in the project delivery case were managers. Even the two interviewed experts were managers of some kind: a project foreman and a chief engineer of planning.

However, by first analysing documentary data and then interviewing not only the supplier’s and the customers’ managers, but also the code generation researchers and their industrial counterparts, I found it relatively easy to identify the evolutionary path of the focal phenomenon: the starting point of code generation research, the emergence of individual skills towards team-based competence, and the fading of the competence in the sense of a remarkably decreased volume of activities. Other facts, such as times, persons, projects, technical results and financial figures could also be traced. Despite, I wish to point out that interpreting an evolutionary process, even when based on explicit data, is a hard problem to solve. At the end of this dissertation I will propose that scientists would be encouraged to carry out action research on industrial networks. This would help to acquire factual data in real time. Furthermore, it would make possible the studying of how the involved persons interpret the data in the view of historical data on events having led to the present situation.

Considering the managerial implications of his research, it would have been interesting if Tikkanen had discussed the emerging process maturity models (cf. [Miller 1995]). The models are frameworks used to assess and improve an organisation's core processes, such as the product development and project management processes of software producing organisations. Because maturity models are based on the idea of comparing the company's processes to the perceived best practices, they could strengthen managerial implications of process studies from the focal company's point of view.

6.1.2 Process-based extensions of the framework

I have included processes into the competence evolution framework by defining certain activities and resources to form a process, and the corresponding focal net to form a governance structure for the process. I have used sequential phases to structure the processes further, and the portfolio of green, blue and red projects to structure the focal nets. Five types of processes have been defined based on the context of contract R&D at VTT, as indicated in Figure 14. Two of them are carried out at the supplier side and two at the purchaser side, so that they form pairs: *competence marketing* (supplier) – *competence purchasing* (customer), and *research and development* (supplier) – *competence exploitation* (customer). At the supplier side, competence marketing aims at establishing projects to carry out the R&D process. At the customer side, the competence purchasing process seeks to exploit the supplier's skills, usually for the development of new products or production processes.

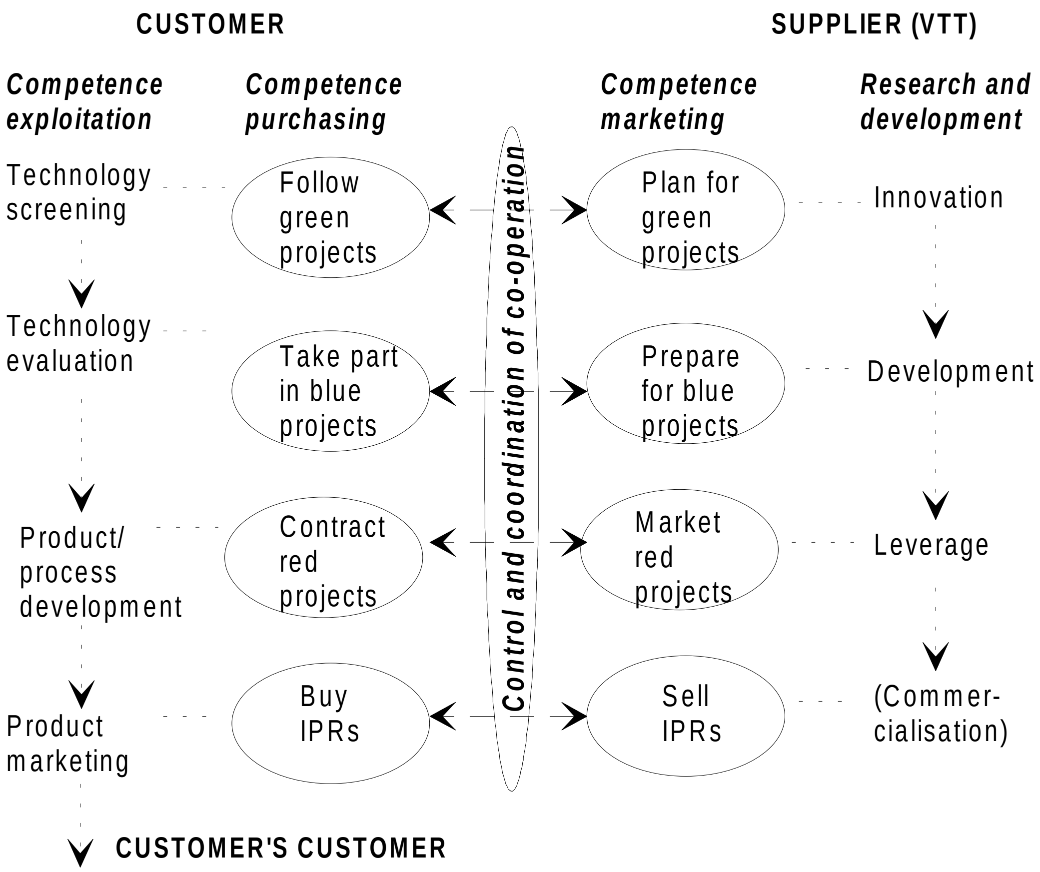

Figure 14. Interplay of the supplier and customer processes.

The processes are associated by the fifth process; co-ordination and control. Phases are defined for each of the four competence related processes, based on the portfolio of green, blue and red projects. At the supplier side, they correspond to the evolution of innovations to commercial solutions, at the purchaser side the change of interest from technology observing to selling of products. Correspondingly, the customers "follow", "take part in", "contract" or "buy" competence from the supplier, which "plans", "prepares", or "markets" projects or "sells" intellectual property rights.

Basically no changes were required because of processes in the focal net part of the framework. Instead of the focal net, the phrase *project net* was taken in use to emphasise the form of focal nets. In one of his two industrial case studies Tikkanen [1998] uses the concepts of focal business process and focal project interchangeably, because “project supplies can also be regarded as meaningful business processes for certain organisations”. Process nets can thus be called project nets, if core activities of the organisation are carried out as projects. In the case of VTT, project nets are usually closely tied with the project nets of customers. Another, less tightly coupled form of interaction is delivery of competence as intellectual property rights (IPR) purchased by customers. This was attempted in the code generation case, as will be discussed in the next chapter.

I will adopt the concept of project nets, pointing out that they may also involve people and organisational units other than members of project teams carrying out the R&D process. For example, higher-level line managers, marketing managers and marketing secretaries may be involved in initiating and planning of new projects, whereas researchers, lower-level line managers and project secretaries may work in the project teams established as a result of such planning activities. The concept of the *project marketing horizon* (cf. Tikkanen [1998]) is also taken in use, to describe the project nets planned to be established by VTT as an R&D supplier organisation. From the customers’ viewpoint, the projects planned to be contracted from suppliers represent the *project purchasing horizon*.

6.1.3 Discussion

In a recent textbook Lowendahl [1997] proposes two dimensions, strategic focus and resource base, for classifying professional service companies, as shown in Table 8. These can be considered as a proposal of a strategic view to focal nets and core business processes, since the former concerns actors and their relationships and the latter the utilisation of resources. One diagonal of the positioning consists of A, B and C types of firms.

Table 8. Positioning of professional service firms [Lowendahl 1997].

Strategic focus/ Resource base	**Client relations**	**Creative problem solving**	**Adaptation of ready solutions**
Organisationally controlled resources	(D) Insufficient adaptability	* → ↓	(B) Efficient
Team-based, individual and collective resources	* → ↓	(C) Both	↑ ← *
Individually controlled resources	(A) Flexible (Effective)	↑ ← *	(E) Lack of co-ordination, discipline

The resource base can be controlled by the organisation as a whole (B), independently by the professionals themselves (A), or as a mixture of the two approaches (C). The choice between A, B and C affects considerably what processes would be conducted by the organisation and how would the processes be carried out. Along the other dimension, strategic focus on customers or customer groups aims at continuing interaction (D), whereas problem-solving based strategies involve a high degree of innovation and solution-based strategies seek to extend markets for uniform services (E). Considerably differing focal nets are required to follow these strategies.

At VTT problem-solving dominates green and blue projects, whereas red projects are either customer driven or solution based. During the competence innovation and development phases, resources are largely controlled by individual researchers. Competence leverage is, on the other hand, most often controlled by the organisation. The competence marketing process follows a similar pattern. Green and blue projects are predominantly initiated and prepared by researchers, whereas line managers are often extensively involved in marketing the red projects. Pressures for the change of a position include, according to Lowendahl, moving away from the "stuck-in-the-middle" position for C types of firms, and moving towards the upper left (D) and lower right corners (E) for B and A types of firms. Contrary to Lowendahl, I believe that it would be beneficial to be stuck in the middle with *core platforms*, to be efficient with regard to competence and effective in managing the portfolio of customer relationships.

6.2 KNOWLEDGE CREATION VERSUS COMPETENCE

Figure 12 at the end of the second part of this dissertation indicates that competence creation and reuse are important dimensions in managing contract research and development. From the viewpoint of competence creation, especially important in the code generation case due to its innovation-related characteristics, competence changes can be relatively straightforwardly associated with the socialisation, externalisation, internalisation and combination functions proposed by Nonaka and Takeuchi [1995], Figure 15.

Tacitness and explicitness of knowledge, i.e. codification/contextuality, constitutes the *epistemological dimension* of the knowledge creation model of the authors. Externalisation that transforms tacit to explicit ("conceptual"/generic) knowledge can be likened to problem solving and the reversed change ("operational"/context-specific knowledge) called internalisation to absorption. The *ontological dimension* of the knowledge creation model can be likened to institutionalisation, as it involves social interaction between the people creating and sharing knowledge. Conversion from tacit to tacit knowledge is called socialisation ("symphatised"/personal knowledge). Explicit knowledge can be turned into other explicit knowledge through combination ("systematic"/organisational knowledge).

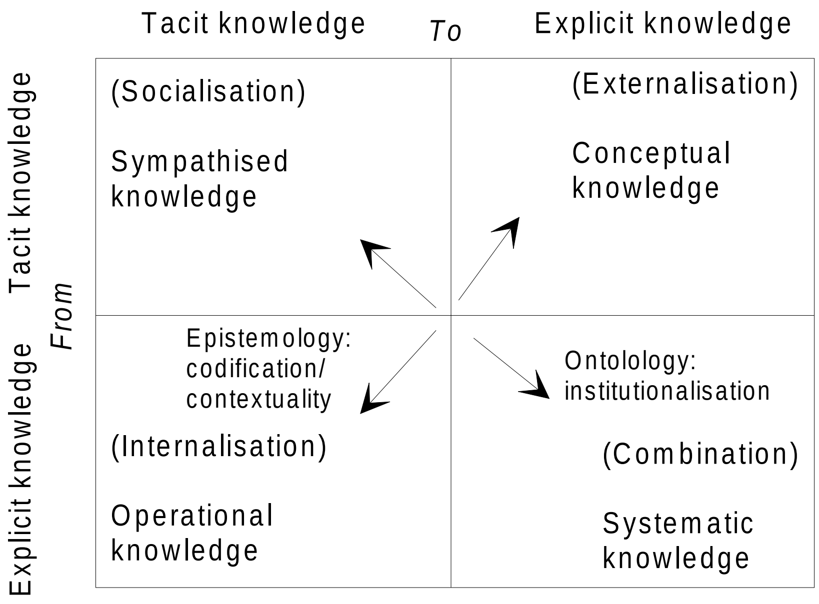

Figure 15. Content of knowledge (cf. [Nonaka and Takeuchi 1995]).

Scanning and diffusion, as they were presented in the initial competence evolution framework, do not make any clear difference between tacitness and explicitness of knowledge with regard to its institutionalisation. However, for example much of the tacit code generation knowledge was not externalised into organisational competence, as opposed to the fault diagnosis case, where an explicit core platform emerged. To be able to study this phenomenon, I will redefine diffusion and scanning as *diffusion by socialisation or combination* and *scanning by socialisation or combination*. They facilitate the associating of the codification/contextuality and institutionalisation of competence, and thereby provide certain side-support to the analysis of different logic of action.

Knowledge creation is interesting also in more general terms. Competence does not evolve only by becoming more explicit. New knowledge must be incorporated continuously to existing skills and capabilities. Nonaka [1994] points this out by stating that organisations must not only process efficiently but also create information. Innovation cannot thus be explained solely as information processing or problem solving, it involves both problem definition and knowledge creation. Nonaka defines knowledge as "justified true belief" and information as "a flow of messages or meanings which might add to, restructure or change knowledge". A theory of knowledge must describe the content of messages, not only their form. Tacitness and explicitness of knowledge constitutes the epistemological dimension of knowledge creation, the ontological dimension involves interaction of people that create and share knowledge.

The three basic factors of the commitment to create knowledge by an individual, a group, an organisation or several interacting organisations are according to Nonaka intention, autonomy and fluctuation. When the four conversion modes between tacit and explicit knowledge are added to this, the "spiral of knowledge" is achieved.

Within an organisation, Nonaka views knowledge creation as consisting of the sub-processes of enlarging individuals' knowledge, sharing tacit knowledge (socialisation), parallel conceptualisation (through externalisation) and crystallisation (through internalisation) of the knowledge, justification for the use of the knowledge within and outside the organisation, and networking of the knowledge to the organisational knowledge base. In [Nonaka and Takeuchi 1995] these five phases are called sharing of tacit knowledge, creating concepts, justifying concepts, building of an archetype and cross-levelling knowledge. These would provide for an alternative to the problem solving, absorption, scanning and diffusion functions that I have applied in this research.

Tacit knowledge can be segmented into two dimensions, technical (know-how) and cognitive (know-what, know what ought to be alias why). The core activity of knowledge creation is "the conversion of tacit knowledge to explicit knowledge", which is related to core competence thinking. However, Nonaka and Takeuchi claim that core competence literature emphasises the behavioural aspects of strategy (how to compete), as opposed to the question of where companies want to compete. Kirjavainen [1997] (pp. 46–52), characterises the view of Nonaka to learning as a process for acquiring, organising and refining knowledge.

There are some fundamental differences between competence-based theories and the process forming a basis for the knowledge creation theory proposed by Nonaka and Takeuchi. First of all, the knowledge creation theory focuses on knowledge explicitly rather than implicitly, concerns knowledge creation as opposed to knowledge processing or management, views middle managers as key "knowledge engineers" in organisations and not as obstacles to competence building, and defines a theory for knowledge creation instead of related but non-coherent principles. The role of middle managers is an interesting issue visible in both of my case studies. In [Nonaka and Takeuchi 1995] the role of middle managers as knowledge engineers is described as follows: "Simply put, knowledge is created by middle managers, who are often leaders of a team or task force, through a spiral of conversion process involving both the top and the front-line employees (i.e., bottom). ... In remaking reality, knowledge engineers take the lead in converting knowledge." A middle manager played a central role in the evolution of the fault diagnosis competence, along the lines described by Nonaka and Takeuchi – not to speak of my personal and other managers' roles in the ultimate developments of code generation matters.

When it comes to practical means for knowledge creation, Nonaka and Takeuchi remain at a rather abstract level. Socialisation can be associated to externalisation by "dialogue", externalisation to combination by "linking of explicit knowledge", combination to internalisation by "learning by doing", and internalisation to socialisation by "field building". For the same reason, it is difficult to say what shared tacit alias sympathised knowledge, externalised alias conceptualised knowledge, combined alias systematic knowledge, and internalised alias operational knowledge actually mean. In this research I had to delve into the changing competence, i.e. in the content of knowledge and the processes shaping it.

6.3 EXPLAINING THE LOGIC OF ACTION

A few quite simple concepts were proposed in Chapter 4 for characterising actors involved in focal nets, e.g. the *role* of an individual either as a manager or a researcher, and an organisation as a research institute or an industrial company. The code generation case indicated various differences in the interests and several conflicts in the viewpoints of different actors. This kind of social embeddedness of networks (cf. [Holmlund and Törnroos 1997]), could not be explained by the simple concept of roles. Miettinen [1993] points out that in innovation-related networks, *voices of actors* should be made explicit for studying, e.g.:

1. different *interests* of the institutions and individuals involved in innovation networks, such as the researchers, product developers and end-users; because of different roles, network participants usually see and interpret each other's interests from different viewpoints.

2. the educational *background*, job history and personal interaction network of individuals, which depend on the views to the roles and positions of the participants of the innovation network, and historically, illustrate how a view has been formed from a person's consideration, opinions and experience from items that originate from various sources and have been affected by other persons' views.

3. the *position* of an individual in his/her organisation, and the division of labour in the organisation; depending on the position of the person, contradictory concepts may be pointed out from the same network, depending in part also on the relations of power, competition and career ambitions of the involved individuals.

Eriksson and Räsänen [1998] use the concept *logic of action* for reasoning about the goals, means and justifications of groups of people. The content of certain logic of action was constructed by the authors from case data, based on the projects and actions each group of actors actually proposed, championed or objected. The case data was also used to make explicit three types of interactions by which groups responded to the actions of other groups over time, called dominance, compromise and integration.

A good characterisation of the concept of logic of action is given in [Bacharach et al. 1996]: "Thus our first assumption is that each party brings to the exchange relationship his or her own specific ends and his or her own specific means for achieving them and that underlying these specific means and ends is some general logic or cognitive framework that guides each party's behaviour. I refer to this framework as a party's logic of action". Bacharach and the others point out the importance of aligning different logic of action, which is one of the central questions in this research: "Our second assumption is that no exchange is possible unless the logics of action of two or more parties involved in the exchange are aligned. In this context, micropolitics may be defined as those strategies and tactics that organisational actors use in negotiating the alignments of logics of action."

However, “the alignment of logics does not mean that there is no conflict between the parties, it simply means that the means and ends of parties across all levels are not inconsistent.” An interesting remark is that if a group shares some logic of action, "it is likely that they will also experience a common sense of dissonance whenever that logic is threatened.”

I will use logic of action to make explicit multivoicedness especially in project nets. I will analyse the logic of action of groups of people involved in certain roles in the nets, such as line managers and focal researchers. The logic of external actors is described only for certain types of organisations, such as funding bodies, other research institutes than VTT and industrial customers. My approach is compared with the one of Eriksson and Räsänen in Table 9. The concepts can be associated to actors, relationships, resources and activities, as is shown in the table and explained below.

Groups of actors with various *objectives* and *goals* to pursue interact within focal nets. Different *forms of interaction* in industrial relationships have been studied by many researchers (cf. [Alajoutsijärvi 1996] as an example). I will use the very basic notions of dominance or submission, co-operation and competition to characterise interaction, in addition to the don’t care situation that denotes a lack of any considerable interaction.

The logic of action of a group can be identified by analysing how its goals are associated with some means, which I will associate with *resource creation, possession* and *mobilisation activities* in the context of focal nets.

Table 9. Making logic of action explicit in project nets.

[Eriksson and Räsänen 1998]	**Project nets**
Manager groups	Groups of certain types of actors
Relationships between managers	Relationships of groups of actors
Form of interaction (dominance, compromise, integration)	Form of interaction in a relationship (co-operation, competition, dominance/submission, don’t care)
Objectives and goals	Objectives and goals of actors
Means	Resource-related activities
Justification of the logic of action	Background, interest and position of actors
Source of influence	Importance (value) of competence, control over certain resources
Confectionery products	Competence
Product mix changes	Evolution of competence
Management of product mixes	Management of competence

The actual content of some logic of action consists of the projects and other activities carried out by a certain group of actors. In [Eriksson and Räsänen 1998] these involve the management of changes of product mixes, in my case the management of competence. The background, interest and network position of actors can be used to justify their logic of action. The *background* of actors in terms of the history of their relationships can be made explicit by a longitudinal analysis of project nets. I will provide such a history for the code generation projects. However, backgrounds include also histories of many other relationships, as well as other developments not involving relationships. For example, the company for which the very first code generation project was carried out, had a long co-operation history with VTT. The R&D manager of the company, responsible for the strategic alliance between the two parties, used to work as a line manager at VTT. I will make explicit these kinds of backgrounds, when it is beneficial to understand the code generation case.

Miettinen [1998] suggests three types of artefacts, which in the context of this research can be likened to product-related resources, in order to characterise the *interests* of actors. I will use his typology here to illustrate interests as a justification of certain logic of action. “What” type of primary artefacts denote tools that are used for carrying out some activities. The VTT code generators are a good example of such artefacts. Secondary “how” type artefacts describe the use of tools in textbooks, journals and papers. A design method developed by the code generation researchers is an example of this: the researchers showed a great deal of interest in transferring the method to industry. “Why” type of secondary artefacts explain reasons for certain phenomena. The focal researchers justified their interest by benefits that would be gained from code generator tools.

Tertiary “where to” artefacts describe the future developments of primary and secondary artefacts, typically as a vision. The software design automation vision presented by the focal researchers is reprinted in Figure 16 to illustrate this. Some people whom I interviewed had a very vivid picture of this vision even after ten years of its presentation.

Another aspect when justifying the logic of action of certain groups of actors is their *position* in focal nets. Johanson and Mattson [1997] define it as follows: “Thus, according to the extended definition, the position of an actor includes also the productive process – in a broad sense – in which it is involved and its direct and indirect network interdependencies. The production role has two dimensions ... The qualitative dimension describes which function the actor has in the production system. ... The quantitative dimension characterises the relative importance that the resources of the actor have in relation to the resources of the other actors, i.e. how much of the total quantity of sustainable resources are controlled by the actor”. The position of a group of actors within a project net can thus be characterised, according to the extended definition, by its activities on certain resources having some relative importance. This is related to the source of influence of a group of actors, which depends on the importance or value of its resources and its possibilities to control their creation and usage.

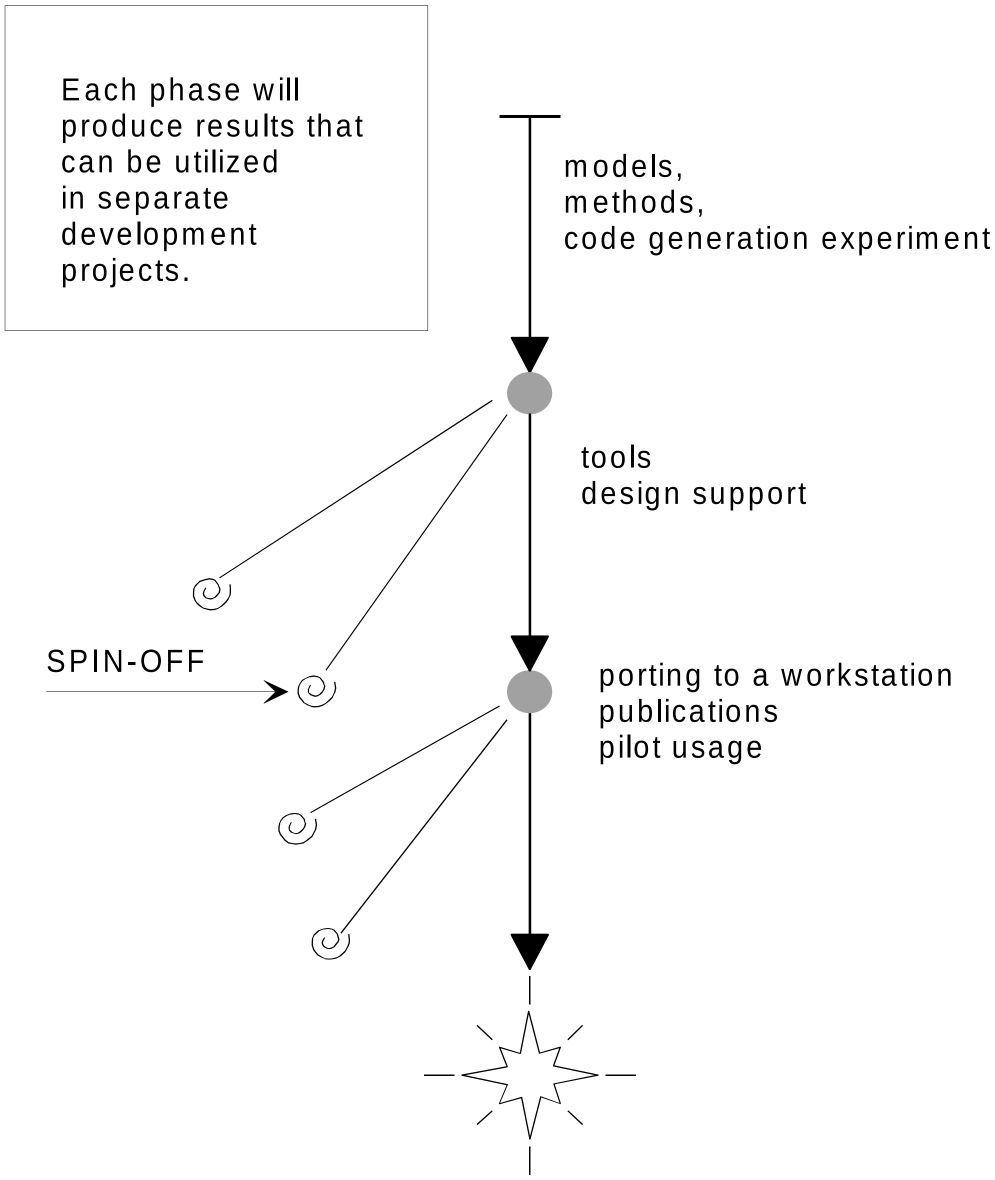

Figure 16. The Sokrates vision, shown as a "guiding star".

Based on the case data, I will view the *value of competence* from the supplier and purchaser viewpoints, using the concepts of *expected* (before), *perceived* (during) and *historical* (afterwards) value or importance.

The expected value of competence is usually documented in various kinds of plans describing how some resources will be developed or utilised, e.g. in contracts, project plans and minutes of the first project management group meeting. The expected value may also be evaluated by some external parties, before the project starts. The perceived value of competence evolves during the project and is documented e.g. in the minutes of project management meetings and in the feedback of the parties or some external evaluators given during or immediately after the work. Customer feedback forms are a routine means to collect opinions on perceived value of competence at VTT. The historical value of competence can be evaluated after some time. In this research the opinions of the interviewed parties indicate the historical value of competence (cf. [Halinen 1994]), after a few or as many as ten years.

One of the best examples of the change of the importance of competence from expected via perceived to historical value is provided by a project in which VTT developed and transferred in use a code generator for a customer company, the firm Nm. The customer's views to the value of the planned and realised results of the project were as follows:

> *Expected value (1993, before the project): "The purpose of the contracted work is to build a real-time software development environment for the MCS project of the customer. The starting point of the project includes the Finsoft/Sokrates technology taken in use and developed further in the RHU project [of the firm Nm]."*
>
> *Perceived value (1994, end of the project):"The specification of the project succeeded well and its implementation corresponds to the specification. VTT's attitude has been positive during the whole project, and problems have been solved fast."*
>
> *Historical value (1998, interview):"Thank you for the questions. I have waited for a possibility to give some feedback. ... Technically, the contribution of VTT was good. From the viewpoint of the business [of the firm Nm] it was fatal."*

6.4 SUMMARY

Figure 17 summarises the main concepts of the revised competence evolution framework, to be used to analyse the code generation case in the next chapter. Notice that the framework does not suggest any specific competence evolution paths, but only associates together concepts that can be used to model such paths. The boldfaced concepts on the left-most side of the figure depict the substance layer: the outer context, groups of actors, relationships, processes that consist of certain phases, and creation and usage of product-related, process-related and organisational resources, which constitute competence. The content of the logic of action of a certain group of actors can be identified by investigating the activities that they have actually been carrying out in the project nets that were realised. Correspondingly, owners of processes and competence can be identified.

I have not presented any explicit model of the outer context and the macro forces that affect competence evolution, as this would have been difficult and somewhat outside the scope of this research. The outer context of the cases, embedded software engineering for the needs of the electronics industry, will be discussed in the next chapter to the extent that is useful to understand the effects of most important macro forces.

The concepts of the dynamic layer of the framework are depicted in the right-most side of the figure. They are the key means to understand and interpret the evolution of the code generation competence, to be discussed in the next chapter. The various kinds of tabular formats applied to make use of the concepts will be explained in detail in the next chapter, since they are used more as representational forms than essential parts of the framework itself.

Outer context ⇐ Macro forces (e.g., economy, technology)

⇑⇓ … *surrounds*

Groups of actors ⇐ Objectives/goals: backgrounds, interests and intra/inter-organisational positions
⇒ Alignment of objectives/goals

⇑⇓ … *who follow certain*

Logic of action

⇑⇓ … *within*

Project nets ⇐ Project marketing and purchasing horizons
⇐ Assuring and balancing of mutuality through
- Forms of interaction
- Scanning and diffusion by socialisation or combination

⇒ Planned versus realised project nets

⇑⇓ … *as the inner context that enact*

Processes
Customer processes: purchasing, exploitation
Supplier processes: marketing, research and development
Co-operation process: control and co-ordination

⇑⇓ … *which form*

Content of logic of action

⇑⇓ … *that consists of*

Activities on resources ⇐ Assuring of capability: life-cycle management
⇐ Balancing of particularity: absorption, problem solving

⇑⇓ … *to create and shape*

Competence ⇒ Evolution/valuation of the competence
- Change of the competence and its value

Figure 17. Overview of the revised competence evolution framework.

7 THE CODE GENERATION CASE

The code generation case will be analysed in this chapter by using the revised competence evolution framework. The analysis is carried out chronologically, based on three main periods that I will call Speco (1985–1987), Sokrates (1988–1991) and Reagenix (1992–1998). They involve the competence innovation, development and exploitation phases of the R&D process carried out by VTT during thirteen years. The case data was organised for analysis (cf. Appendix 2 of [Seppänen et al. 1999a]) around two closely related processes of VTT as an R&D supplier: competence marketing, with a focus on the project marketing horizon, and competence building in the form of research and development. Competence purchasing and exploitation processes were addressed at the customer side. The four processes were associated together by a fifth process, the control and co-ordination of co-operation, as was shown in Figure 14 in Chapter 6.

I will first analyse the outer context of the case and then proceed to the inner context consisting of the actually realised projects. Six groups of key actors involved in the case are then addressed: the managers of the focal organisation, the focal code generation researchers, their colleagues, industrial customers, research customers and Tekes. The goals and logic of action of these actors, which affected the code generation project nets and processes and thereby the code generation competence, are analysed. The actual project nets are compared with the planned nets, i.e. the intended project marketing horizon. The project nets are also compared with the project purchasing horizon, to the extent that the intended purchasing of the code generation competence could be made explicit from the case data. The nets are structured according to the phases of the R&D process.

The forms and means of interaction of the parties within the project nets are first addressed by evaluating the balancing and assuring of the mutuality of VTT's activities, based on diffusion and scanning. Then, the content of the logic of action is discussed by evaluating the balancing of particularity and assuring of capability of code generation resources, based on life-cycle management, problem solving and absorption. Finally, the evolution and valuation of competence are analysed, concerning product-related, process-related and organisational competence. Finally, the results of the analysis are summarised from the viewpoint of building and exploiting competence.

7.1 INTRODUCTION

In the code generation case VTT managers aimed at creating a growing portfolio of fully contractual red projects, involving in particular machine automation companies. The focal researchers favoured marketing generic code generator techniques and tools. The customers of VTT had difficulties in coping with these two logic of action, in a rapidly and radically changing business environment. This may have been the reason for the competence, in the end, being utilised mainly by VTT itself. This did neither benefit its developers nor advance the evolution of the competence.

The differing logic of action of the two key parties, which resulted in the lack of any considerable portfolio of external customer relationships, lead to a rather rapid withering of the competence at VTT. However, the developed code generation technology has recently been sold to the former researchers, who have established a company based on their own logic of action. This kind of competence survival through many years and despite conflicting viewpoints is, after all, a key factor in making business out of research.

Early eighties – emerging industrial interest

Industrial software development emerged as early as in the sixties. The phrase software engineering was taken in use in the seventies to denote a systematic approach to organising, managing and performing software design as part of product development. The need for embedded systems software development originates also from the seventies, when microprocessors became commercially available. At first, embedded software was mainly developed by hardware designers.

In the early eighties, VTT Computer technology laboratory (TKO) was one of the main contributors in Finland to bringing embedded software engineering principles, methods and tools to the attention of industry. It even proposed and made popular the Finnish translation of the term embedded software, “sulautettu ohjelmisto”. TKO’s persons took part in the development of a number of very successful electronic products, as embedded software engineering experts. Moreover, the roots of some early Finnish software development tools are to be found at TKO.

Embedded software engineering research started to flourish in Finland in the mid-eighties. Tekes, the national technology agency, organised two quite extensive software-related research programs. Companies that had started to utilise microprocessors realised that the size of software incorporated in their products was rapidly growing and that they needed better means for managing software development. They also found out that the features used for arousing the customers' interest in products were often based on software. Industry was, for these reasons, quite eager to join the steering groups of public research projects dealing with software. Only modest fees, compared to the total budgets of the projects, needed to be paid for joining the groups.

At the same time, industry started to hire experts to facilitate the use and development of new software engineering methods and tools. Quite many of the experts had previously been working at universities and at VTT, which would contribute to industrial interest in public research projects. While product development engineers were playing a major role, other people from industry, such as marketing and customer support professionals, were seldom involved. Most of the representatives in funding bodies, such as Tekes, were technology experts as well.

Turn of the nineties – joint national research

Tekes established the first national software-related research program called Finprit in the mid-eighties. TKO carried out several Finprit projects and produced prototypes of embedded software development tools. The second software engineering research program, called Finsoft, was launched by Tekes in the late eighties [Karjalainen 1991]. TKO prepared and carried out the biggest project within the Finsoft program, called Sokrates. Over ten companies took part in this project. I will use acronyms to denote these and other firms involved in the code generation case, in order to preserve anonymity, required by some of the interviewed persons. Anonymity is also justified by the fact that the case includes several controversial aspects, as was pointed out in Chapter 2.

The aim of this project was to produce a design method and to build a *code generator* prototype, which would provide for a systematic, seamless and automated embedded software development approach. The general goal of the project was characterised as creating *design automation* for embedded systems. The intention was to turn embedded software development from art to an engineering discipline, relying on rigorous methods and tools.

Conflicting viewpoints arose around code generation at VTT as early as in the early nineties. One of the very basic controversies involved line managers and focal researchers. The conflict was related to the basic R&D strategy of VTT based on a portfolio of green, blue and red projects.

As shown on the left-hand side of Figure 18, code generation research started "backwards", in the red Speco project paid by the firm K, a key customer of TKO in the area of software engineering. The subsequent Sokrates project was a very large blue project, in which VTT's own financing was quite limited, due to the as high as 70% funding by Tekes and the fees paid by the companies taking part in the project.

Although the expectations for a number of subsequent red projects were high, after Sokrates both public and industrial investments on code generation related activities decreased. Four rather small red projects were carried out for three industrial members of the Sokrates steering group, and two additional projects for other companies.

Despite the efforts allocated to the project, neither a breakthrough in code generation succeeded, nor did any vendor commercialise the code generator prototype developed in the project. The rights to the code generator have recently been sold by VTT to a small company owned by some of the original researchers. Therefore, there still is a possibility for the venture to succeed. However, in this research I am mainly interested in investigating why the code generation competence could not be more extensively exploited earlier, to allow VTT to make use of its research results.

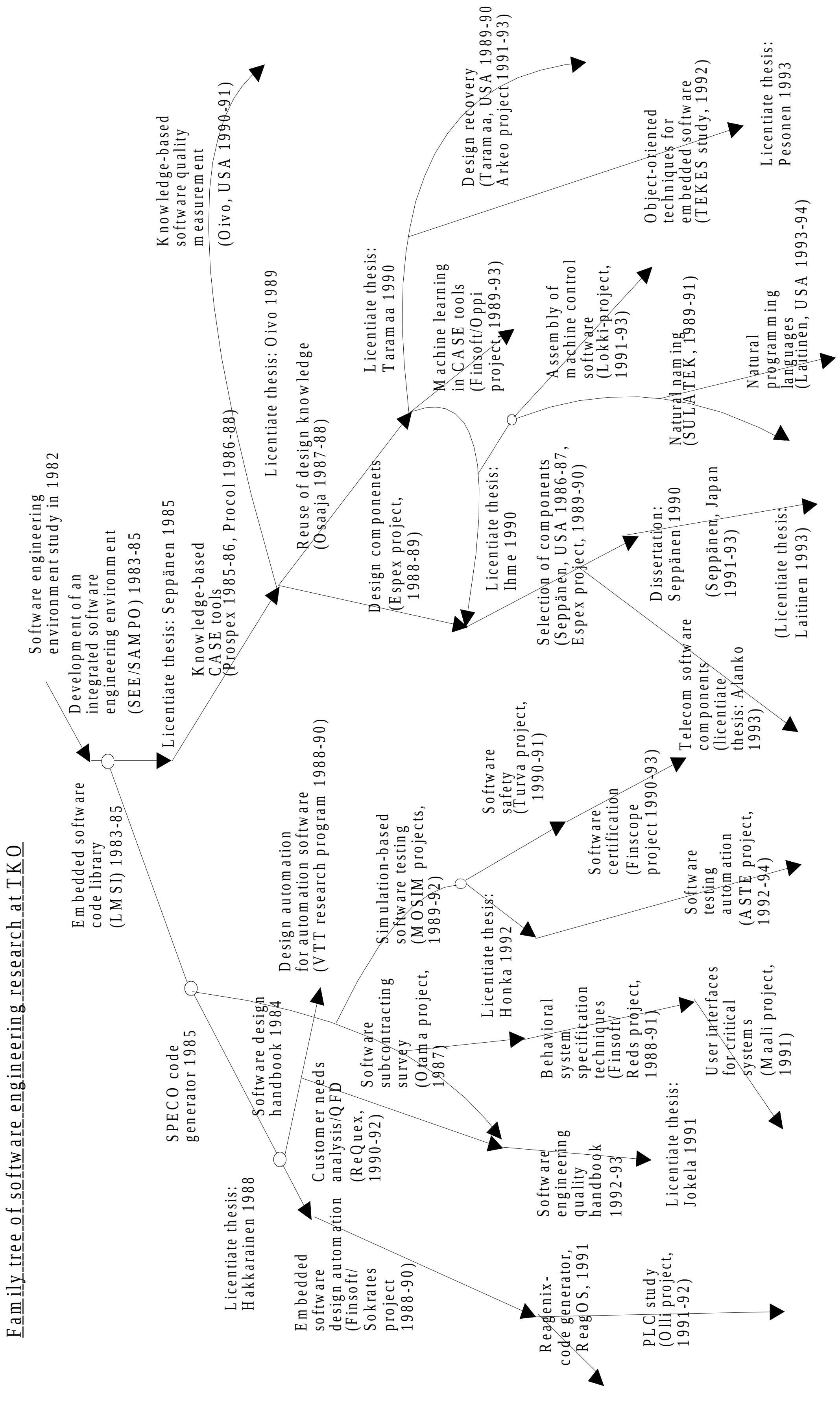

Figure 18. Software engineering research family tree at TKO 1985–1993.

Late nineties – declining R&D activities

Although some licenses of the code generator were sold to industry, the generator was mostly used in blue projects at VTT. On one hand, the code generation researchers felt that managers were preventing them from making a practical tool out of the research results. They were not allowed to continue their work by any considerable internal funding, while the managers were hoping to establish red projects paid by industrial customers. On the other hand, the managers' patience for expanding industrial income from red projects was gradually exhausted:

> *Code generation researcher: "The management of TKO said that this [the code generator] is not good, this is a prototype, a hack-hack, and we cannot seriously offer this to industry. Now, having been in industry for four years, I have seen what kind of tools are being used, for example in the telecommunication area: tools developed by ourselves and by universities, public domain tools, and our competitors' tools. I must say that Reagenix was, after all, a very reasonable tool. The impression that was given by the management of TKO was that industry uses only top-quality, well-packaged tools offered by reliable parties. This is the point, VTT did not see itself as a reliable tool developer."*

> *Manager: "Well, it might be that there was a belief that something would come out, but that belief gradually vanished. In closing, [of the Sokrates project] ... [the focal researchers] took it for granted that about six companies would take the results in use right away. They already wrote down ready-made contracts, but the fact was that there were no real deals at all. They told ... that everything would have been agreed, but the reality was completely different. We lost confidence in their judgement."*

As an organisation, VTT gave up code generation as an R&D topic quite soon after the blue Sokrates research project had been finished in 1991. Figure 19 taken from the strategic plan of the focal Software engineering section does not include code generation in the "develop systematically" phase of software engineering technologies. The SA/SD (called "SA/RT" in the figure) methods and CASE tools, which were utilised as part of the code generation techniques developed by VTT, are considered to belong to the "give up" life-cycle phase. Code generation has not yet become a notable approach to embedded systems development in more general terms either.

However, one third of the industrial representatives participating in a global survey concerning the strategic needs and future trends of embedded software technologies carried out in 1996 [Seppänen et al. 1996] indicated that code generation techniques will be needed in the future. At the time of writing this dissertation, VTT has subcontracted expert work from the small company established by the former code generation researchers, to conduct a strategic assessment of code generation technologies for one of its key customers. It is already confirmed that the customer will use some code generator in the development of software embedded into its future products.

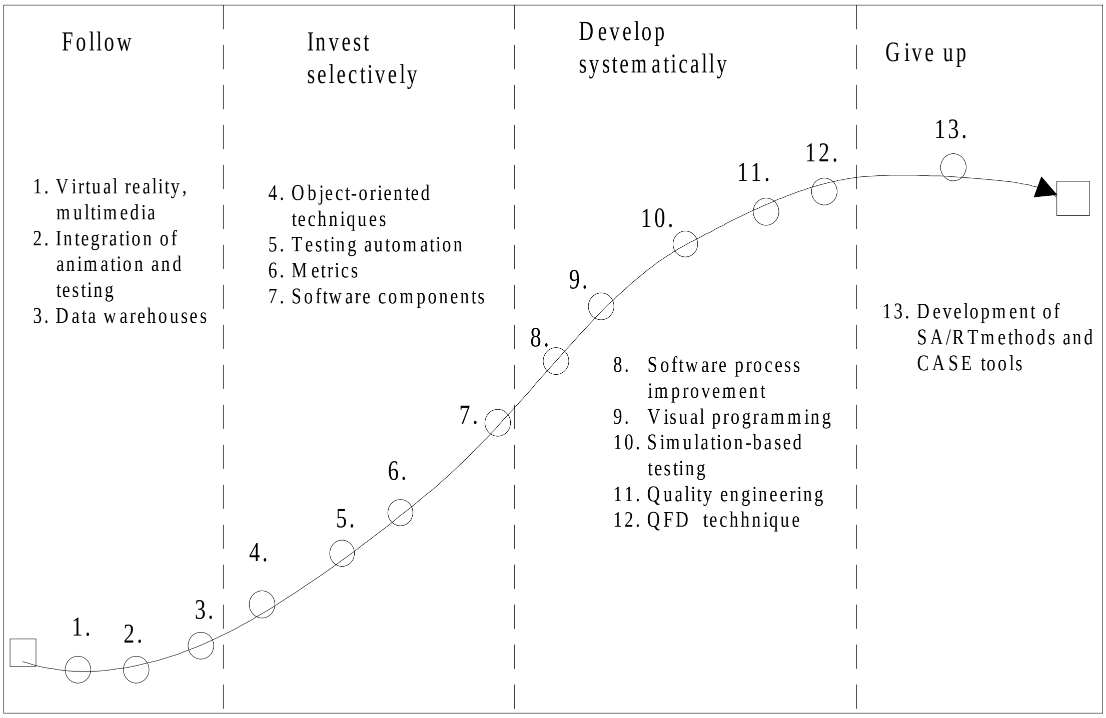

Figure 19. Software engineering maturity life cycle (as of 1993).

7.2 MACRO FORCES OF THE OUTER CONTEXT

Tikkanen and Alajoutsijärvi [1998] define the outer context of industrial relationships as “an extension of the connected network directly relevant to a customer-supplier relationship and its inner context”, where the connected network focuses on “whole organisations as collective actors”. The context in which the code generation projects of VTT were conducted can be characterised as *industrial embedded software engineering*.

As indicated above, the developments started in the seventies, with microprocessors becoming incorporated in products of a different type than general-purpose computers. Software development was at first carried out to a large extent as a side-job of hardware design. By the mid-eighties it was emerging as a new profession. As part of the development, Tekes started to actively support national research and development of embedded software engineering in Finland. The firm Is started the Finnish business on embedded software engineering methods and CASE tools in 1985.

I will first outline changes in the industries developing and applying embedded software in Finland, focusing on the electronics industry. The case data is used for discussing the position of the focal organisation in this context. The technological changes which have shaped industrial embedded software engineering are then summarised. The role of the focal organisation in regard to these changes is briefly evaluated. Frame 8 lists the data sources I have used in tracing the changes related to industrial embedded software engineering from 1985 to 1998.

Anon. 1998. Huippuosaajat maailmalla. Sähkö- ja elektroniikkateollisuusliitto. 21 p. (in Finnish)

Hienonen, R. 1997. Elektroniikka- ja sähköalan kehitysnäkymät 1997–2002. VTT Automation. 229 p. (in Finnish)

Anon. 1996. Teknologia 2000. Osaamisella tulevaisuuteen. Tekes. 120 p. (in Finnish)

Anon. 1994a. Suomi tietoyhteiskunnaksi – kansalliset linjaukset. Ministry of Financing. 73 p. (in Finnish)

Anon. 1994b. Osaamisstrategia. Sähkö- ja elektroniikkateollisuuden menestystekijät, avainteknologiat ja osaamisen kehitystarpeet. Seteli. 29 p. (in Finnish)

Lovio, R. 1993. Evolution of firm communities in new industries. Acta Academiae Oeconomicae Helsingiensis. Series A:92. Helsinki School of Economics and Business Administration. 304 p.

Anon. 1993a. Teknologiakatsaus 1993. Tekes. 77 p. (in Finnish)

Anon. 1993b. Kansallinen teollisuusstrategia. Kauppa- ja teollisuusministeriön julkaisuja 1/1993. Ministry of Trade and Industry. 124 p. + app. (in Finnish)

Kivisaari, S., Lovio, R. 1993. Suomen elektroniikkateollisuuden merkittävien innovatiivisten liiketoimintojen menestyminen 1986–1992. VTT Technology Research Group. 51 p. (in Finnish)

Anon. 1992a. Teollisuuspoliittinen linjaus. Seteli. 16 p. (in Finnish)

Anon. 1992b. Sähkö- ja elekteroniikkateollisuus. Seteli. 7 p. (in Finnish)

Anon. 1990a. Valtion tiede- ja teknologianeuvosto: katsaus 1990. Tiede- ja teknologiapolitiikan suuntaviivat 1990-luvulla. 76 p. (in Finnish)

Anon. 1990b. Teknologiaohjelmatoiminnan linjat 1990-luvulle. Komiteamietintö 1990:2. 112 p. + app. (in Finnish)

Anon. 1989. Oulun läänin teknologiapoliittinen ohjelma. Oulun lääninhallitus. 54 p. (in Finnish)

Anon. 1987a. Uuden teknologian tuotekehitys- ja tutkimuskeskus, perustamisselvitys. Teknologian diffuusio, TEDI-87. Projektiraportti n:o 1. City of Oulu. (in Finnish)

Anon. 1987b. Uusia keinoja teknologiayhteistyön tehostamiseksi Oulun seudulla. Teknologian diffuusio, TEDI-87. Projektiraportti n:o 2. City of Oulu. (in Finnish)

Anon. 1987c. Pohjois-Pohjanmaan elinkeinoelämän tietotekniset toimintaedellytykset. Pohjois-Pohjanmaan Seutukaavaliitto. Publications Series A:84. 61 p. (in Finnish)

Klus, J.P., Markkula, M., Venho, J., Järvenpää, A., Sirkeinen, U., Ahlroos, R. 1985. Effective technology transfer. 128 p.

Mårtenson, G., Otala, M., Wiio, O.A. 1985. Tietotekniikka 1990-luvulla. Sitra. Series B, No. 78. 112 p. (in Finnish)

Frame 8. Sources of data on the outer context from 1985 to 1998.

A considerable hardship for this analysis was caused by the fact that embedded software can be incorporated in almost any kind of electronic products. No studies have yet been carried out on the role of embedded software in even the best-known Finnish electronic products. The nineteen data sources listed in the frame provide thus only an overview of the subject. One of the most thorough analyses of the evolution of the Finnish electronics industry is given in [Lovio 1993]. However, the study only concerns events to the year 1989. I have also used [Kivisaari and Lovio 1993, Mårtenson et al. 1985] and a few reports of the local activities in the Oulu region to provide some insights in the industrial context at the time code generation related activities were initiated at TKO. Hannu Hakalahti, the director of TKO, was involved in carrying out some of the local studies.

Several national information technology studies and strategic plans are available from the early nineties. I selected the ones acquired by the managers of the focal organisation and used by them to support strategic organisational planning. Later on in the nineties, both Tekes and the Association of the Finnish Electronics and Electrical Engineering Industries have published several surveys and forecasts, as indicated in Frame 8. Embedded software was, however, included as a distinct section in the forecasts as late as 1995 by Jukka Karjalainen, the former vice laboratory director of TKO. The corresponding section in [Hienonen 1997] which I have used in this research was authored by myself.

7.2.1 Economy – Finnish electronics industry 1985–1998

Vision – towards intelligent machines

A famous Finnish technology expert Matti Otala wrote the chapter “Information technology in products” in the forecast [Mårtenson et al. 1985]. He chose microprocessors used in cars as an example of embedded systems, but also pointed out other automation applications, such as elevators, cranes, heavy-duty machines, public and industrial transportation equipment, as well as robots. The section “Software production” (pp. 60–61) indicates that embedded software would be applied especially in machines and equipment and developed “semi-automatically” by using special software engineering environments based in part on “artificial intelligence techniques”.

Otala was expecting the automation in the development of embedded software to increase “slower than [in] the design of other electronics [technologies]”, three to four times by 1990. This was to a great extent the view of the firm K, when it contracted the initial code generation project Speco from TKO in 1985. In reality, embedded software did become crucial in telecommunication, but only ten years later. The size of software in such products increased by the factor of four in 1994–1998, the productivity of software design only by the factor of two.

Reality – automation and telecommunication as key clusters

[Lovio 1993] analyses the growth of the Finnish electronics industry from 1961 to 1989. After a remarkable decrease in growth in 1981, the level of 15% annual increase was reached in 1985. After that, the growth stagnated again and was well under ten percent in 1989. Lovio views the period between 1975 and 1989 a *competence enhancing phase* of the Finnish electronics industry, comprising the *era of ferment* based on micro-electronics from 1975 to 1984 and the *era of incremental change* from 1985 to 1989, with "new products in all old product groups" being introduced, and both exports and internationalisation increasing.

Considering the companies involved in the code generation case, N, Nm, Nr and Nt as parts of a larger corporation and the firm V were among the top five, when ranked by the number of employees in 1985. The firm K was among the top ten and the firm W among the top twenty. In 1989, the top five companies had remained the same, but the firm K had dropped down as many as nineteen positions. The firm W had risen among the top fifteen and the firm Th among the top twenty.

Kivisaari and Lovio [1993] analyse "innovative businesses" of the Finnish electronics industry between 1986 and 1992, including some of the companies involved in the code generation case:

- in the product group "industrial automation and measurement devices", the firms K and V are "large innovative companies",
  - the annual turnover of the *automation sector* was FIM 4.8 billion in 1992, (14% of the total production), *decrease of 4%* from 1986,

- in the product group "telecommunication equipment", the firms N and Nt are "large companies",
  - the annual turnover of the *telecommunication sector* in 1992 was FIM 7.2 billion (as much as 43% of the total production*),* *increase of 6%* from 1986,

- in the product group "consumer electronics", the firm S is a "large company", and
  - the annual turnover of *consumer electronics* was FIM 3.2 billion in 1992 (only 6% of the total production), *decrease of 14%* from 1986,

- in the product group "medical electronics", the firm W is a "medium-size company".
  - the annual turnover of *medical electronics* was FIM 1.2 billion in 1992 (only 2% of the total production), *decrease of 2%* from 1986.

The firm Nm had grown by 25% between 1986 and 1992, the firm Nt by 21% and even the firm K by 17%. The turnover of the firm S had increased by 9% and that of the firm W by 5%. The turnover of the firm V had decreased by 4% and it had laid off 900 employees.

TKO had three key customers at the time of the preparation of the biggest code generation project Sokrates in 1987:

- *firm K* was seen as a “lowering customer” in terms of the income form the co-operation, which had increased greatly from FIM 500 000 in 1984 to 2.3 million in 1985, but then decreased to 1.9 million in 1987, and

- *firm N,* where the income from the co-operation had increased rapidly from FIM 500 000 in 1983 to 1.5 million in 1984, but then settled at the level of 1 million by 1987, and

- *Tekes,* whose financing for TKO had increased from FIM 0.8 million in 1984 to 1.5 million in 1985 and to 3.2 million in 1987.

The total volume of the industrial income of TKO had increased from 4.9 million in 1984 to 6.8 million in 1985 and to 7.4 million in 1987. It was thus more than twice as much as the income TKO received from Tekes.

TKO – investing in automation applications

Two years later in 1989 TKO had about 25 industrial customers. The industrial income was approximately FIM 6 million, i.e. it had started to decrease due to the extensive involvement of the institute in Tekes-funded research. The estimated total of annual R&D driven subcontracting volume in the electronics industry was 120 man-years. In the machine and equipment manufacturing industry the estimated annual embedded systems development volume was about 500 man-years. Nevertheless the volume was annually increasing by 25%, especially in software development. However, R&D driven subcontracting was estimated at only 25 man-years annually. The share of TKO of the total annual R&D subcontracting market of the two sectors, 145 man-years, was approximately 14% in 1989.

Two years later, in 1991, when the Sokrates project was finished, 37% of the financing of TKO came from public funding bodies, mainly from Tekes. Another 37% came from industry. The situation had thus changed considerably from the one of 1987. The number of the industrial customers of the laboratory had remained the same, but in addition to some fifteen electronics companies TKO had as many as ten machine and equipment manufacturing companies as customers. In 1987 it had had three contractual customers from the machine automation industry.

Comparing the customer related data of TKO from the mid-eighties to the early nineties with the data on the evolution of the electronics industry, it is likely that the increase of the telecommunication sector was not foreseen – granting that the increase was missed also by many others. The increase of the co-operation with the firm N was not comparable to the 25% annual increase of its turnover. As an example, co-operation concerning the company’s in-house code generator did not materialise. The fall of the consumer electronics segment had been rather dramatic between 1986 and 1992, as much as 14%. However, the firm S, for example, was still considered an important customer for code generation by TKO.

One of the strategic visions followed by TKO at the turn of the nineties was that computers would be embedded in machines and industrial equipment, as envisioned already in the eighties by Matti Otala, among others. However, according to [Klus et al. 1985] the change of the importance of electronics and real-time production control technologies in the metal industry, as seen by executives, had been estimated to remain less than 3 on the scale 1–4. In the contrary case, the importance of software technologies in the electronics industry had been estimated to increase from 3 to 3,5 by both executives and engineers. This kind of an evolution had indeed begun already in 1989. In contrast, the annual R&D subcontracting value of the machine and equipment industry was estimated to be only one fifth of the corresponding value of the electronics industry in 1989. Still, TKO had increased the volume of its machine and equipment manufacturing customers to as high as 40% of its industrial customer base by 1992.

There was also a clear difference in public funding for electronics and information technology research, compared with machine automation [Anon. 1990b]. Tekes had spent FIM 70 million in addition to the other organisations 45 million on the Finprit program during the period 1984–1988. For the subsequent Finsoft program the corresponding figures were 40 and 15 million between 1988 and 1990, when the program was ongoing. The total sum of the two programs was well over FIM 150 million. The funding of Tekes for the mechatronics research program between 1987 and 1989 was only FIM 20 million, and an additional FIM 10 million was received from other sources. This was altogether less than one fifth of the two software-related research programs.

At the end of the eighties VTT had established a Machine automation laboratory in Tampere. The laboratory was utilised for researching and developing machine control solutions based in part on electronics and embedded systems. The laboratory had the advantages of being located at the heart of the Finnish machine industry and employing a considerable number of automation and control system experts. The VTT Electronics laboratory had also been carrying out R&D on mechatronics for several years. [Anon. 1987a] includes a piece of news (Insinööriuutiset, 13.4.1987) titled “VTT [Electronics laboratory in] Oulu leads research. Electronics to machine parts according to the conditions of mechatronics”. Mechatronics is mentioned as one of the foci in the Oulu region in [Anon. 1987c], which states that “the Electronics laboratory of VTT located in Oulu carries out research that falls in part in the area of information technology”.

Telecommunication – miraculous success despite the recession

The overall situation in the outer context during the birth of the Reagenix code generator in 1991 was such that TKO was extensively involved in an industrial segment that was small, had increased quite slowly and was not keen on using external R&D services. Moreover, there were competing offerings for this segment even inside VTT. In comparison, the electronics industry had rapidly increased (in the Oulu region even faster than elsewhere in Finland [Anon. 1989]), was using external R&D services more extensively and had started to consider software as a core technology.

In addition, TKO had become largely dependent on external income, not only from the machine industry, but also from Tekes, when a very deep economic recession hit Finland at the turn of the nineties. VTT was not only competing against the commercial code generator launched by the firm Is during the deepest recession, but the tool was also marketed especially to the machine and equipment industry and to consumer electronics and instrumentation companies suffering considerably from the recession. [Anon. 1993b] points out that "many indicators show that the technological development of industry [in Finland] seems to have become slower. Investments in research have increased less than earlier. The increase of exports rested almost entirely on a few large companies. The role of new innovative small enterprises has been too small".

The telecommunication sector was estimated to "consist of a functional whole which has yet only weak cluster structures", but it was "strongly oriented towards exports and has increased rapidly also during the recession" at the beginning of the nineties:

- in 1991 the value of the production of the electronics industry was FIM 12 billion, of which:
  - 24% came from telecommunication,
  - 11% from industrial automation and instrumentation, and
  - 6% from consumer electronics

- in 1992, the value of the production had increased to FIM 15 billion, of which
  - 29% came from telecommunication,
  - 9% from automation and instruments, and
  - 4% from consumer electronics.

By 1997, the technology review 1993 of Tekes estimates that the foci of development in automation include "hardware and software engineering", but a weakness is "the lack of companies that develop production automation systems" [Anon. 1993a]. Concerning telecommunications, it was estimated that "the great importance and hard competition in the field mean that Finland cannot be a forerunner in critical issues. Big competitors can take in use standards that differ from the ones used by the Finnish companies." Of course this vision was thoroughly false.

According to the competence strategy of the electronics and electrical engineering industries written only a year later in 1994, development of automation systems "is based strongly on the familiarity with the sector[s] using the systems"[Anon 1994b]. "System design tools have become a central part of product design, because the efficiency and competitiveness of products depend critically on … tools". However, the telecommunication sector had "reached the World Class level … and become involved in the creation of new standards". In addition, the Tekes technology review written in 1996 states that Finnish "companies must be involved in affecting and specifying new standards for the field" [Anon. 1996].

Furthermore, the “most important technical competence areas” of the telecommunication sector are presented to be “embedded real-time software, signal processing, ASIC design, radio engineering, telecommunication systems and system-level software”. The review includes for the first time a special section on embedded software (pp. 31–32). Between 1994 and 1997, 24 000 new jobs had been created in the electronics and electrical engineering industries, in 1997 44% percent of all new employees held a Master’s degree in engineering and 33% an engineering college degree. During 1989–1993 the labour of the electronics and electrical engineering industries decreased to 34 400 persons, but already in 1996 it had risen to 51 000 persons [Hienonen 1997].

In 1996 the value of the production of the electronics and electrical engineering industries was FIM 54 billion, of which the share of:

- the electronics industry was 41 billion,
- the telecommunication sector 46%,
  - the increase of production had been over 30% in 1996
- the automation sector 7%, and
  - the increase of production had been approximately 15% in 1996
- the consumer electronics sector 2%.
  - the production had decreased by 20% in 1996.

About 31% of the personnel of the electronics industry was involved in R&D in 1996, but in the telecommunication sector as much as 47%. In 1987 the corresponding values had been 19% and 28%. In industrial automation and instrumentation, the volume of R&D personnel was 15% in 1987 and 16% in 1996, in consumer electronics the volumes were 7% and 14%. [Hienonen 1997] includes an estimate of the volume of personnel in electronics industry involved in software development. The estimate shows that the telecommunication sector employed over 30 times more software engineers in 1996 than the second biggest sector, industrial automation and instrumentation. In 1997 the value of the R&D spending in the electronics industry was almost FIM 17 billion, 61% of the total national R&D spending [Anon. 1998]. “Software plays[ed] a key role” in the electronics industry, and almost one fifth of the personnel of electronics and electrical engineering companies was carrying out software development – in telecommunication companies as much as one fourth.

VTT – restructuring after the recession

VTT was restructured in 1994, presumably, at least in part, due to the recession. The former small independent laboratories were integrated into nine large units. The mechatronics section of the Electronics laboratory was incorporated with the Machine automation laboratory and some other laboratories with VTT Automation. The Computer technology laboratory became a part of VTT Electronics, together with the rest of the former Electronics laboratory and two other laboratories.

The hardware engineering related research groups of the new unit ELE started to specialise in telecommunication electronics. The software engineering groups had a wider technology portfolio and customer base, but were also deeply involved in the R&D of embedded telecommunication software and the methods and tools used in software development. By 1998, the annual volume of embedded software R&D at ELE had risen to over 70 man-years, of which almost 40% involved confidential customer projects. 50 to 70% of these projects were carried out for telecommunication companies. The annual size of the biggest project portfolio contracted by a single customer was about 15 man-years. It included activities concerning software method and tool development, as well as technology transfer.

The machine and equipment industry had become one of the key sectors for fault diagnosis and intelligent systems R&D at ELE [Seppänen et al. 1998a], but not for R&D on embedded software engineering. Only occasional customers had been gained from the consumer electronics sector. Comparing the dramatic increase of the telecommunication sector between 1993 and 1997, also in terms of the number of software engineers, with the evolution of the other sectors, it is very likely that there could have been a strategic market for the code generator. The right door for stepping into the rapidly moving telecommunication sector might have been offered as early as in 1991, or during 1992–1993 at latest. However, at that time TKO was busily further developing the code generation technologies for the needs of machine and equipment manufacturers (e.g. the firm Nm), as well as for the consumer electronics sector (e.g. the firm S).

7.2.2 Technology – embedded software engineering

Tables 10 and 11 taken from [Seppänen et al. 1996] present the technological developments in embedded software engineering in the nineties, from the viewpoints of system solutions and design methods and tools. The road maps shown in the tables are based on a survey conducted by the authors of the report including some 150 professionals around the world. The purpose of the survey project, managed by myself, was to support the planning of embedded software related R&D activities in an electronics research program launched by Tekes in the late nineties.

Table 10 indicates that embedded software solutions have evolved in the nineties from small, dedicated device control programs to large, heterogeneous and networked software systems. At the same time, many special in-house solutions have been replaced by commercial or at the fewest standardised solutions. Interfaces that did not exist or were “closed” for internal use only, have become publicly defined and “open”. Embedded software solutions have been integrated with new hardware solutions, such as ASICs, to form deeply embedded systems.

Table 10. Overview of an embedded software solutions road map.

Road map area	**1991**	**1996**	**2001**
System characteristics	Device Control	Mass customisation, portable products	Networked systems
User interfaces	Alpha-numeric	Graphical, customisable	Virtual, COTS
System software and hardware	In-house software and COTS hardware	PC-compatible software and hardware vs. ASIC-based co-designs	Open system platforms vs. COTS designs
Data access, management	Dedicated local data storage	Data management Systems	Real-time networked multimedia
Associated IT systems	Local information processing	Client-server system interconnections	Embedded and IT system networks
Communication infrastructure	Closed, local area	Open local area and Global networks	Heterogeneous, Wireless

From the business perspective, these developments mean that the earlier focus on special-purpose technological solutions has been re-directed to applications of electronic products and services provided by the solutions. Many companies use standard software-hardware platforms and concentrate on delivering customer value for certain applications. Others still compete by using special technological solutions if the markets are large enough or if there is a need either for protecting company-specific technological innovations or developing one-of-a-kind systems, such as space devices.

Considering the system solutions developed in the Sokrates project, such as a communication protocol package and an operating system kernel, the developments would have required focusing either on special solutions or accommodating to some standards. The use of the operating system kernel solution by the firm N and the communication protocol package in a space instrument represent the former. No serious attempts were made regarding the latter. An example could have been provided by developing a software bus concept starting from the protocol package, nevertheless aiming at a solution that would be compatible with an emerging object-oriented communication standard. The evolution of embedded software development methods and tools has led from structured techniques, such as SA/SD, first to object-oriented and then to component-based techniques (Table 11).

One of the main reasons for the development is the increase in the use of standard software and hardware platforms and open system interfaces. Software engineering environments have changed from isolated and site-specific to heterogeneous and inter-operating support systems that sustain a process-based viewpoint to software production. A good example of this is provided by the Capability Maturity Model (CMM), which is an incremental process improvement framework, as opposed to the “quantum leap” in industrial software engineering envisioned in the Sokrates project.

Table 11. Overview of an embedded software process road map.

Road map area	**1991**	**1996**	**2001**
Development methods	Structured development, CMM/1	Early object-oriented development, CMM/2	Component-based development, CMM/3–4
Development tools	Structured CASE tools	Object-oriented CASE tools	Application-specific and simulation CASE tools
Environments	Isolated environments	Site-specific environments	Interoperable local & global environments

The process perspective has not resulted in such an integration of methods and tools as was forecasted in Sokrates. Instead, for example the CMM model was built mostly by software quality professionals, more interested in the management of software development than in specific methods and tools. The evolution of embedded software engineering technologies shown in the two tables has been extended and reformulated in Table 12, by using the elements of software engineering competence. The periods of *art*, *craft*, *standstill* and *engineering* shown in the table illustrate changes in embedded software development practices in Finland. These phases have been defined by myself for the needs of this research, based on the information discussed earlier in this chapter. In the mid-eighties, the field involved few hundred hardware-oriented system engineers. In the late nineties, thousands of professionals work with embedded software, possibly involving millions of lines of program code.

I have indicated *dominating elements* of each period, if there are any, in italics in the table, *competing elements* as ordinary text and emerging *substitutive elements* in parentheses. These characterisations are based not only on information about the outer context, but also on the code generation case data, where e.g. the substitution of the SA/SD design technique with object-oriented (OO) techniques is clearly visible. I will indicate the direction of industrial interest in certain embedded software applications by using four symbols: rapid growth (“>>”), growth (“>”), steady state (“+/–“) and decrease (“<”). For example, device control applications prevailing earlier, have given way to telecommunication applications with regard to interest. Automation applications have grown, but considerably lesser than expected in the eighties. Applications of electronic instruments have remained the same or decreased in the nineties.

It is worth noticing that there has been no considerable growth in the Finnish CASE tool business either, although the firm Is, for example, has played a key role in providing embedded system design methods and tools for industry. Many companies have given up the building of in-house methods and tools, with an exception of some large telecommunication companies. However, in the view of the table, and as pointed out by many of the interviewees of this research, the Sokrates project was carried out at the right time. The goal was to transform the craft of program design to a software engineering discipline.

Table 12. Elements of embedded software engineering during 1985–1998.

Period	Applications	Techniques	Technologies	Functions
Art (–1985)	>> Control	*Program design* (SA/SD)	*Compilers Debuggers*	*Coding, Testing*
Craft (1986 –1991)	> Control > Automation > Telecom +/– Instruments +/– EDA	*SA/SD* Program design (Co-design) (OO design) (Testing)	*Programming Environments* (CASE tools)	*System design,* System testing, Documentation
Stand-still (1992 –1993)	+/– Control > Automation > Telecom +/– Instruments +/– EDA	–"–	*CASE tools* Visual programming environments	–"– (Quality engineering)
Engin-eering (1994 –1998)	< Control > Automation >> Telecom +/– Instruments +/– EDA +/– Multimedia	*OO design* SA design? Co-design (Simulation) (SDL) (TTCN)	Visual CASE OO CASE (System design tools) (ASIC tools) (Web tools)	(Simulation) *Multi-paradigm design,* *Concurrent engineering*

7.3 GROUPS OF ACTORS INVOLVED IN PROJECT NETS

On the basis of Figure 17, I will start the analysis of the code generation case from the inner context of project nets. The objectives, goals and logic of action of the six groups of actors involved in the code generation case are made explicit. The code generation project nets are then compared with the project marketing and purchasing horizons.

7.3.1 Objectives, goals and logic of action

Table 13 shows the goals and Table 14 the logic of action of the groups of actors involved in the code generation, as they were identified in the case data. Four of the six groups of actors are further classified in the table:

- VTT managers into directors and sections heads (i.e. high-level general managers and low-level technical managers),
- Code generation researchers into two teams, called Sokrates and Reagenix according to the names of two generators,
- Industrial customers into the Sokrates project participants, broad co-operators, focused buyers and tool vendors, and
- Research customers into the focal organisation itself and other research organisations.

Table 13. Goals of actor groups between 1985 and 1998.

Groups	Actors	Goals of actor groups		
		Speco 1985–1987	Sokrates 1988–1991	Reagenix 1992–1998
A: VTT Managers	Directors Section heads	Serve firm K Nurture R&D projects	Follow research Market new red projects	Don't care Control human resources
B: Focal Resear-chers	Sokrates team Reagenix team	Implement technological Innovations	Carry out world-class research in a different way	Create new type of product business at VTT
C: Funding Bodies	(KTM) Tekes	Finance the screening of technological limits	Finance generic research and aid in the transfer of technologies	Finance applied research based on explicit industrial needs
D: Industrial Customers	Sokrates parties Broad co-ops Tool vendors Focused buyers	– – – Try new technologies	Follow ongoing novel research Gain new ideas Consider using the technology Try new technologies	– Buy solutions Be alerted Buy certain new technologies
E: Research Customers	TKO/ELE Other institutes	– –	Make use of research results	Solve design-related problems cheaper or faster
F: Colleagues	VTT research scientists	Discuss on technologies	Argue about goals and results	Use in projects, if not too risky

Alignment of the objectives and goals

One of the main objectives of the *VTT managers* was to increase the volume of contract R&D, while minimising financial risks and earning small profits as a national government-owned research organisation. The *focal researchers* wished to make a paradigm shift in the design of embedded software, by building and transferring in use innovative design methods and tools. In principle, the goals of the groups should have been well-aligned, actually similar. Nevertheless, they appeared to be quite diverse.

Industrial customers were keen on following and evaluating new technologies, but mainly by making rather moderate investments and taking only limited technical risks. The sooner this kind of technology screening would result in direct business benefits, the better. *Research customers* were aiming at certain project goals not usually related to code generation. They were looking for free or inexpensive use of effective technologies, which tended to irritate the focal researchers. The research customers and VTT managers did not recognise this, though.

The strategy of *Tekes* involved starting and co-ordinating national research initiatives as a funding body, to "investigate the limits of new technologies", as the strategy was described by one of the interviewed Tekes representatives. *VTT colleagues*, mostly informally related to the focal researchers, also had their own professional and personal objectives.

Table 14. Logic of code generation action during 1985–1998.

Groups of actors	**Objectives and goals**	**Means**	**Justifications**	**Logic of action**
A: VTT managers – research directors – section heads	Increase of the volume of contract R&D: minimise risks, earn small profit	Control of human and organisational resources	Collective R&D: Increase volume as decision makers	*Contractual R&D* – project business – resource control
B: Focal researchers – Sokrates project team – Reagenix team	Make a paradigm shift: transfer in use new design methods and tools	Control of project resources, personal technical skills, team-based problem solving	Small-scale R&D: Innovation as technology experts	*Small entrepreneurship* – pioneering research – product business
C: Tekes	Guide national research initiatives: investigate limits of technologies	Control of financial resources, program-level co-ordination	National success: Act as a "patron"	*Investments in the future*
D: Industrial customers – Sokrates parties – broad co-operators – focused buyers – tool vendors	Follow-up and evaluate new technologies: make moderate investments with small risks for the benefit of the business	Freedom to accept or reject the developed technology, business and application skills	Application focus: Early technology screening/adoption as development managers or experts	*Technology testing* – smooth audience – paradigm buying – solution buying – technology watch
E: Research customers – within TKO/ELE – other organisations	Aim at certain project goals: utilise effective technologies for free	Control of project resources, use of project relationships	Project-based R&D: Project goals as project managers	*Problem solving* – cheaper solutions – faster solutions
F: Colleagues	Ensure professional and personal careers: carry out applied engineering R&D	Personal skills, project relationships, (money, time)	Applied research: Personal goals as "observers"	*Professional interest*

Overview of the logic of action

The logic of action of a group of actors does not merely combine of the goals and means of the group. It also includes the justifications and arguments supporting the viewpoints of the group, based on the backgrounds, interests and positions of the actors.

The logic of action of the *VTT managers* has been described in the table as "Contractual R&D". This means the project business based on the control of human and organisational resources. The *focal researchers* emphasised "Small entrepreneurship", in the form of pioneering research during the Speco and Sokrates periods and product-based business during the Reagenix period. The logic of action of the *industrial customers* can be characterised as "Technology testing". The Sokrates steering group was a suitable audience for innovative research. Broad industrial co-operators wished to exploit comprehensive solutions, whereas focused buyers were interested in specific technologies. Tool vendors followed the work during and immediately after Sokrates, while not actually participating in the work. The logic of action of the *research customers* was "Problem solving", by using inexpensive or faster technical solutions.

Tekes wished to make "Investments in the future", by financing R&D of new technologies. *VTT colleagues* expressed "Professional interest" in the work, but were not directly involved.

Focal researchers

The focal researchers, whose professional background involved rather small-scale R&D, were interested in making innovations as technology experts, not as project or organisational managers. Technological innovations, as understood by the focal researchers, were in the Sokrates project produced by conducting world-class research in a different way from all the other projects of TKO. The different approach caused problems with the VTT managers and colleagues.

During the Reagenix period, a related novel aim was to create product-based business at VTT. For this reason the goals of the VTT managers started to differentiate remarkably from the goals of the focal researchers by the mid-nineties.

Appendix 2 illustrates some of the roots of the logic. The researchers were stressing in the interviews that some of them had "run real companies" before joining VTT. However, e.g. the complaints of the Sokrates steering group regarding the preparation and delivery of plans, status reports and documents point towards the logic of small-scale business: paperwork was done after the more interesting technical questions had been answered.

Still, the logic was quite appropriate during Sokrates. According to Tekes, the Finsoft evaluators and the steering group, the project succeeded in reaching its goals. The disagreements between the focal researchers and some of their colleagues regarding the way of carrying out innovative research were not considered a problem either.

The logic run into difficulties during the Reagenix period. The focal researchers were then left with practically no project resources at all at their disposal, while the managers who controlled the process-related and organisational resources continued to follow their own logic. However, it was as late as 1998 when the managers did sell rights to the code generation technology. One may well ask, why did this not happen earlier. One possible answer is that the generator was used in quite a number of blue projects during the whole of the nineties. A more likely reason is, however, the project-based business logic of the managers. They were neither keen on selling licenses, nor of making money by selling the IPRs of VTT for good.

VTT managers

The background of VTT managers can be characterised as collective R&D, as opposed to focused individual specialists. One of their main interests was to increase the volume of R&D as organisational decision makers. They focused on launching and nurturing projects. As an example, they took to a great extent the side of the key customer, the firm K, in a dispute over the Speco project. However, only after project-based exploitation of the Sokrates results had in their view failed, started their goals to differentiate from the goals of the focal researchers. They were not any more actively marketing red code generation projects in the late nineties, even though they did not oppose the use of Reagenix in blue projects.

The driving force behind the logic of action of the VTT managers was the goal to increase the volume of R&D. As an example, TKO grew from about twenty to almost a hundred people in ten years, during the period 1983–1993. The close to non-profit nature of the services provided by VTT meant, according to the logic of the managers, that green and blue research topics should move to the red portion of the project portfolio after quite a short while. In other words, industrial customers should have covered the costs of the refinement of capabilities built in green and blue projects.

Profits from red projects have been allowed at VTT only since the mid-nineties. For example, the main red code generation project MCS-REA was offered to the firm Nm at a prime cost, with no profit margin. Later during the nineties, the profits gained from Reagenix licenses appeared to be very small, no considerable red code generation projects were carried out and the tool was used for free in joint blue projects managed by VTT. The project business logic of the VTT managers was actually most effective during the Sokrates period, when TKO was earning the highest external income from code generation R&D.

In the interviews, the focal researchers and some of their colleagues were criticising the unwillingness of the VTT managers to take risks, be they financial or technical. On the other hand, the managers of the firm Nm described the consequences of serious risks that had been realised. The focal researchers still had the original belief that taking the risks of technology testing had paid off for the firm Nm. The customer’s managers were characterised as intelligent people. In the interviews they blamed in part themselves for the realisation of code generation related risks.

The focal researchers should have, in principle, followed the same logic of action as project managers and researchers did. However, this was not the case. The early dispute with the firm K during the Speco project had a lot in common with the disagreement with the VTT managers later. The manager of the firm K wanted to have well-established schedules and plans, while the innovators were solving problems at a hectic pace in the basement.

Customers

The background of many of the *industrial customers* involved in the code generation case revealed expertise in certain applications rather than in industrial embedded software engineering. Like the firm K already in the late eighties, they were interested in early screening and evaluation of new technologies, mostly as product development managers and engineers. After the Sokrates project, focused buyers acquired certain technologies from VTT, whereas broad co-operators looked for more comprehensive solutions.

After all, on the basis of my interviews, these goals were not taken into consideration either by the VTT managers or the focal researchers. The firm Nm, a broad co-operator very disappointed with the results of collaboration, is a good example. The lack of co-operation with the firm N, developing its own code generator and involved in an enormously successful business, is another example. No co-operation emerged with tool vendor companies either – the business goals of the CASE tool developer firm Is, which took part in the Sokrates project, clashed with the goals of the focal researchers later in the nineties.

Research customers were interested in achieving certain project goals, often as project managers. They wanted to make use of Sokrates and Reagenix results to solve various kinds of problems, for the benefit of the industrial partners of their projects. Some of them criticised the VTT managers for not sufficiently attending to the code generator. Still all customers were quite satisfied with the support received from the focal researchers. They did not recognise problems regarding the free support.

Colleagues

The VTT colleagues, not utilising the results, were acting as observers rather than active participants during Sokrates and especially later. For this reason they did not have notable influence on the developments.

Tekes

Tekes, the funding body, was acting as a kind of patron in projects. Its interest was to ensure national success in developing and exploiting new technologies. However, Tekes redirected its funding in the nineties from generic research, such as the Finsoft program, to projects with more direct links to certain industrial needs. Tekes did not provide any direct funding for code generation related work after 1994.

The goals and logic of action of Tekes, industrial customers and research customers were quite well aligned with each other during the Sokrates period, when the logic of the focal researchers was prevailing. Even after Sokrates, until 1993, the logic of Tekes, the VTT managers and the focal researchers did not conflict, since they were all wishing to make use of the results of Sokrates. However, due to the industrial logic of testing new technologies, as well as to the recession, only a few rather small exploitation projects emerged. The firm Is introduced its code generator, which many of the companies employed instead of Reagenix.

This can be understood as part of the emerging Engineering period in the development of embedded software. During the earlier Craft and Art periods industry was more willing to follow and test research results. The whole Finsoft program is a good example of this trend. Industry, Tekes and the VTT managers were looking for more needs-driven software engineering research after the Standstill period 1992–1993. Such new topics as animation and unit testing used to renew the several years old code generation vision could not compete with the new visions of object-oriented and component-based software engineering. The Reagenix code generator became an in-house tool for VTT to be used in blue research projects. The interest of these projects was not to carry out or pay for any code generation research. However, the logic of action of the focal researchers seems to be occupied with such an idea.

7.3.2 Planned versus realised project nets

Project marketing and purchasing horizons

The project marketing and purchasing horizons are compared with the realised project nets in Table 15. During the Speco period, embedded software engineering was still considered art. However, it started to emerge as a distinct profession for which more effective tools and methods were needed. Tekes had been established in 1983 and TKO was in many respects in a pivotal position for launching, carrying out and co-ordinating national research on embedded software engineering. For this reason, the focus of the code generation project marketing horizon of TKO involved joint blue projects. This was also the main project purchasing horizon for industry and Tekes. The ongoing phase of the code generation R&D process was innovation rather than technology development. Correspondingly, the phase of the competence exploitation process was technology screening.

The actual colour of the Draco code generation pre-study project carried out in 1987 can be considered to be green, although KTM was involved as an external funding body. The earlier Speco project was red, involving both technology evaluation and early development. There was thus a mismatch between the project marketing and purchasing horizons and the realised project nets. The managerial problems encountered in the Speco project can be seen as one result of the mismatch: the researchers aimed at innovations, the managers expected to get value for the money they had spent (the firm K) or earned (TKO).

VTT had not yet expressed any considerable interest in internal purchasing of embedded software engineering technologies. No code generation project nets had thus been established within VTT. During the Sokrates period embedded software engineering evolved from art to craft, but the period ended with a standstill caused by the economic recession. Industry was eager to take new methods and tools in use in product development, but the recession would decrease its interest towards the end of the period. The survival, or rather the miraculous growth, of the telecommunication sector was not yet foreseen. The ongoing phase of the R&D process was technology development. The project marketing horizon of the VTT managers included both red spin-off projects of Sokrates and blue projects, in which code generation could have been applied. The focal researchers were also looking for opportunities for selling tool licenses.

The competence exploitation phase related to the project purchasing horizon was technology evaluation: Tekes was expecting to see industrial spin-off projects, industry was also interested in the base techniques of code generation, such as the SA/SD design technique. The VTT Electronics laboratory was ready to exploit the early Sokrates results first in blue and then in red projects. The actual project nets fitted well with the project marketing and purchasing horizons, although they involved not only technology evaluation but also some technology screening. Sokrates, Synchro, Sasic and Sixa created together a considerable portfolio of blue projects including two VTT laboratories, Tekes and a large number of companies. Kaapeli was the kind of a red spin-off project that VTT and Tekes wished to launch already when Sokrates was ongoing. After the Sokrates project had been finished and the Reagenix period started, the standstill hit in 1992–1993 and only gradually turned to an actual embedded software engineering era. The Finnish industry, led by the telecommunication sector, was recovering from the recession.

The traditional consumer electronics sector had nearly disappeared and the importance of the automation industry had decreased in comparison with the electronics industry. Industrial interest in developing in-house methods and tools had faded, since international tool vendors were well represented in Finland, not to speak of the firm Is and some other national tool vendor companies. The telecommunication sector needed new methods and tools for managing large software systems, but were looking mostly for commercial solutions. The software process had emerged as an organisational rather than a tool-based view on software development.

The phase of the code generation R&D process expected by the VTT managers and Tekes was the one of leverage. The focal researchers accompanied this with a "commercialisation" phase to be carried out by VTT itself, after the attempted co-operation with the firm Is had failed. The project marketing horizon of the VTT managers included red projects, targeted especially at machine automation. VTT was considered an internal customer. The managers were also aiming at establishing blue projects, to apply and further develop code generation techniques, including joint European research projects. The focal researchers focused on selling licenses, to gain money to conduct the further development of the generator.

Table 15. Planned and realised project nets during 1985–1998.

Code generation vs. industrial software engineering periods	Project marketing horizon (vs. phase)	Project purchasing horizon (vs. phase)	Realised project nets
Speco/Art…Craft: Embedded software engineering started to emerge as a distinct profession, tools and methods were needed	***Innovation*** Blue embedded systems projects: joint software engineering research	***Screening*** *Tekes*: blue projects *Industry:* blue projects *VTT*: --	***Evaluation*** Speco: red research Draco: pseudo-blue (green) research
*Sokrates/*Craft…Standstill: New software engineering methods and tools had been developed and taken in use, but the recession decreased industrial interest	***Development*** *Managers*: red spin-off or blue research projects *Researchers*: red spin-off projects, red licenses (blue and green projects)	***Evaluation*** *Tekes*: spin-off projects *Industry*: tool piloting and system design projects, SA/SD courses *VTT*: blue/red projects	***Screening and evaluation*** Sokrates: blue research Kaapeli: red spin-off Synchro, Sasic, Sixa: blue research
*Reagenix/*Standstill…Engineering: Industry was recovering from the recession led by the telecommunication sector, but its interest in in-house general-purpose methods and tools had decreased; new methods and tools were sought for managing large systems; the software process emerged as an integrative view on software development	***Leverage*** *Managers*: red projects especially in machine automation, blue projects (incl. joint European projects) ***"Commercialisation"*** *Researchers*: internal and external licenses, green tool development projects, red generator-based software development projects	***Process and product development*** *Industry*: well-packaged effective methods/tools *Tool vendors*: buying of IPRs, elimination of likely competitors *VTT*: well-packaged and supported in-house (cheap/effective) tools *Research institutes*: problem solving means	***Process and product development*** Reagenix, Aniprosa: green research: generator-based red projects, license selling: usage in blue research

The code generation competence was in the phase where it could be exploited in industry, along with the internal research customers at VTT looking for well-packaged and effective embedded software engineering methods and tools. Commercial tool vendors were interested in preserving and extending their markets, including buying of IPRs and elimination of competitors. The realised code generation project nets, shown in Table 15, included small-scale green projects. A few red code generation projects were launched. The main project, called MCS-REA, was carried out for the firm Nm. Some tool licenses were also sold.

Table 16 summarises the code generation project nets, from the viewpoint of external partners involved in the projects. Blue project nets dominate, green nets are almost missing and only few red nets were created from the many blue nets. VTT Electronics laboratory established the kinds of red machine automation related projects nets at which TKO aimed in the early nineties, but the generator did not turn out to be any strategic technology for them either. The recession finished the broad co-operation between TKO and the firm Nm. The relationship with the firm K expanded from the red Speco project to the joint blue Sokrates project came also to an end. There was hardly any continuation in project relationships with focused buyers.

Table 16 shows that during the period 1985–1998 there were altogether three green projects, one blue project, in which code generation competence was developed, nine others, in which it was exploited, and nine red project nets managed by the focal organisation. In three projects, the blue Sokrates project net consisting of more than ten firms was specialised in red customer project nets. The firms N and S in these nets were focused buyers and the firm Nm a broad co-operator. Not only code generation related co-operation but also other co-operation with most of the other Sokrates parties finished entirely. The firm Is remained one of the commercial tool vendors of VTT. None of the blue projects where the VTT code generators were exploited resulted in red projects of the focal organisation. However, tool licenses were sold to companies participating in the blue projects.

The firms R and Nc and VTT Food technology laboratory represent customers with whom red project nets were established without any previous participation in blue code generation projects. The relationships with them involved single projects, none of them have contracted any other projects afterwards. The firm El did not become any partner of VTT in the further development of the code generator.

The longest lasting customer relationships, six years, involved the original industrial research partner K and the key industrial exploiter Nm. The former lasted from 1985 to 1991, the latter from 1988 to 1993. Figure 20 shows how the number of green, blue and red project nets changed during 1985–1998, without considering the parties involved in the projects.

Table 16. Code generation project nets 1985–1998.

Project partners	**Project nets**			**Remarks**
	Speco (1985–1987)	Sokrates (1988–1991)	Reagenix (1992–1998)	
KTM	Draco: 1987	--	--	Pseudo-blue
Firm K	Speco, Speco-2: 1985–1987	Sokrates: 1988–1991	--	Red-to-blue (expansion)
Firm N	--	Sokrates 1988–1991	Osdyn: 1992, Mots-2: 1997	Blue-to-red (specialisation, focused buyer)
Firm E, W, So, P, Nt, Pa, Ta, Th, T, V	--	Sokrates: 1988–1991	--	Collaboration with the firms T and V continued
Firm Is	--	Sokrates: 1988–1991	(Tool vendor in 1992–1998)	VTT was a focused buyer
Licence buying firms	--	--	Reagenix: 1992–1997	Focused buyers
Firm El	--	--	Aniprosa, Aniprosa-2: 1993–1994	Subcontracting from the firm El by VTT
Firm Nm	--	Sokrates: 1988–1991 Kaapeli: 1990–1991	MCS-REA: 1993	Blue-to-red (specialisation, broad co-op)
VTT Electronics Laboratory + firms	--	Synchro, Sasic: 1989–1990, Tulko: 1989–1993	--	Broad co-operator
Firm R	--	--	Raski: 1992	Broad co-operator
Firm S	--	Sokrates: 1988–1991	Sympa: 1992–1993 Cute: 1993	Blue-to-red (specialisation, focused buyer)
VTT Food technol. lab.	--	--	Kaasu: 1992	Focused internal buyer
Firm Kc	--	--	Nosto, Nosto-2: 1993–1996	Focused buyer
Finnish Work Environment Fund + firm T	--	Turva: 1991–1992	Diag: 1992–1993	Sokrates-SA method used (blue/no colour)
Finnish Work Environment Fund + firm H	--	--	Rulla, Rulla-2: 1993–1996	Reagenix code generator used (blue)
University of Oulu + firms	--	Tulko: 1989–1991	Rekki: 1995–1996	–"– (blue/no colour)
STUK + VTT Automation	--	--	AVV: 1995–1997	–"– (blue/no colour)
Tekes	(funding firm K/Speco project)	Sokrates: 1988–1991 Sixa: 1990	(funding blue projects)	Funding Kaapeli and MCS-REA projects

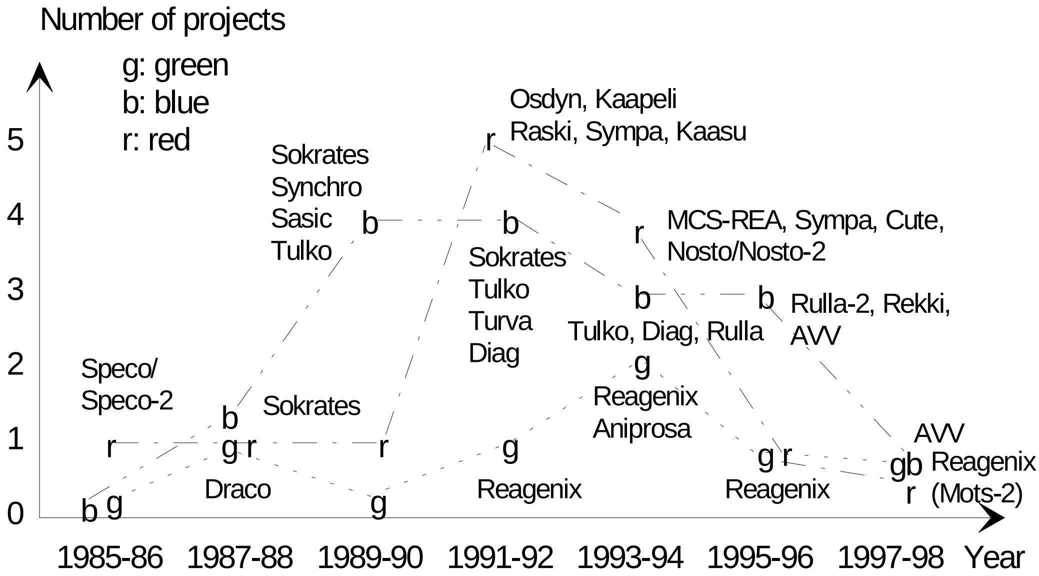

Figure 20. Change of the number of projects during 1985–1998.

From the viewpoint of customer relationships this meant the following:

- one red relationship with a key customer (the firm K, machine automation) was expanded to a blue network,
- one blue network with a key customer (the firm N, telecommunication) was specialised in a red relationship,
- two blue networks with new customers (the firm S, consumer electronics; the firm Nm, machine automation) were specialised in red relationships,
- eight out of the eleven other customer relationships of the blue Sokrates network finished (the firms Nt, V and T were former customers, the rest were new customers),
- the blue network with the firm Is changed to a series of tool purchasing transactions (from the firm Is to VTT),
- three new red customer relationships were created (the firm R, consumer electronics; the firm Kc, machine automation; VTT/ELI; research institute/scientific instruments),
- nine blue (in part colourless from the viewpoint of the focal researchers) code generation exploitation projects were created, involving three different funding bodies, several research partners and a large number of industrial firms,
- one green relationship was created (the firm El), and
- a number of red license selling transactions took place.

This is summarised in the following as a chronological list of twenty three projects during the Speco, Sokrates and Reagenix periods, in order to prepare for the subsequent analysis of the change of these project nets:

Speco (1985–1987)

Speco (1985–1987, red): firm K, TKO

Draco (1987, green/pseudo-blue): KTM, TKO

Sokrates (1988–1991)

Sokrates (1988–1991, blue): Tekes, TKO, firm K, firm E, firm W, firm So, firm P, firm Nt, firm Pa, firm Ta, firm Th, firm T, firm V, firm Is, firm S

Synchro, Sasic (1989–1990, blue): TKO, VTT Electronics laboratory, Tekes, University of Oulu, several firms

Tulko (1989–1993, blue): TKO, VTT Electronics laboratory, Tekes, University of Oulu, several firms

Sixa (1990, blue): TKO, Tekes, a few other research and industrial parties

Kaapeli (1990–1991, red): TKO, firm Nm, (Tekes)

Turva (1991–1992, blue/no colour): TKO, Work Environment Fund, firm T

Reagenix (1992–1998)

Diag (1992–1993, blue/no colour): TKO, Work Environment Fund, firm T

Osdyn (1992, red): TKO, firm N

Reagenix (1992–1997, red): TKO/ELE, several license buying firms

Raski (1992, red): TKO, firm R

Sympa (1992, red): TKO, firm S

Kaasu (1992, red): TKO, VTT Food technology laboratory (VTT/ELI)

Cute (1993, red): TKO, firm S

MCS-REA (1993, red): ELE, firm Nm, (Tekes)

Aniprosa, Aniprosa-2 (1993–1994, green): ELE, firm El

Nosto, Nosto-2 (1993–1996, red): ELE, firm Kc

Rulla, Rulla-2 (1993–1996, blue/no colour): TKO/ELE, Work Environment Fund, firm H

Rekki (1995–1996 blue/no-colour): ELE, University of Oulu, a few firms

AVV (1995–1996 blue/no-colour): ELE, VTT Automation, STUK

Mots (1997, red): ELE, firm N.

The four years from 1991 to 1994, after the Sokrates project, were the time of the most extensive networking. Figure 20 shows how at the time the relative number of green, blue and red projects nets followed the ideal project portfolio. However, most of the blue projects were exploiting already existing results, instead of solving new code generation problems. As the focal researchers stated in the interviews, this did not bring much income for them to carry out further research and development on code generation. Quite the opposite, it often meant extra work with no income.

If the blue exploitation projects were removed from the portfolio, actually only the Sokrates project would remain. In other words, there was no continuation at all in the joint code generation research, a rather typical situation in more general terms: public funding for a specific applied research topic may last for two to three years, but after that the topic should be considerably revised or integrated into other topics to ensure continuing funding. This did not succeed for Sokrates. It was attempted, once, after a few weeks the massive Sokrates project had been finished, as a very small part of another research topic, and in the middle of the recession.

The number of red code generation project nets increased rapidly at the end of the Sokrates project, despite the standstill caused by the recession. However, their number also decreased rapidly. In about five years the number of project nets was back to the level of one project from which it started in 1985. The number of green project nets was very low, when compared with the blue and red project nets. However, Figure 20 provides a clear illustration of the fact that before Sokrates started in 1988, the number of green projects increased according to the ideal project portfolio principle.

The development of the Reagenix code generator after the Sokrates project was, on the other hand, carried out mainly in red projects. The number of green projects again increased in 1992–1994, when the number of red and blue projects had started to fall. However, this could not prevent the falling. The blue project nets decreased more slowly than the number of red project nets, but they also ended up at the level of one project in 1997–1998.

7.3.2.1 Mutuality – forms of interaction

The forms of interaction between the focal researchers and the other groups of actors are illustrated in Table 17. The processes owned and the resources controlled by the actors are also shown, in order to associate the discussion with the contents of logic of action addressed in the next section. The most influential groups, on the basis of the resources they created or possessed as is shown in the table, are indicated by using *italics*, rather influential as ordinary text and less influential in parentheses. Non-influential groups of actors are also listed and reasons for the lack of their influence outlined.

During the Speco period the most influential group of actors was the focal researchers, due to their having the individual and team-based problem-solving skills for carrying out the innovative research needed for realising the code generation vision.

Table 17. Interaction of code generation parties in 1985–1998.

Period	Influence of actor groups	Forms of interaction	Core processes	Key resources
Speco (1985–1987)	*Researchers* VTT managers Firm K	Co-operation Co-operation Dominance	R&D (innovation) Project marketing Control, R&D	Individual and team-based problem-solving skills, vision Contacts with Tekes/industry, project planning skills Money, decision making power, technical skills, company vision
Non-influential: KTM		**Reasons**: KTM did not want to control technology evaluation in Draco, but just finance the project.		
Sokrates (1988–1991)	*Researchers* (Tekes) (Industrial customers)	Dominance Submission/ Co-operation Co-operation	R&D (dev.) Evaluation Control Project purchasing, Exploitation	Sokrates project resources, vision Money, program-level resources Money, system design and application skills
Non-influential: VTT managers, Research customers, Sokrates parties, Colleagues		**Reasons**: VTT managers, research customers and Sokrates parties did not control the R&D process, only co-ordinated it. Colleagues could not even co-ordinate the process, only comment its results.		
Reagenix (1992–1998)	*Research customers* VTT managers (Researchers) (Tool vendors)	Co-operation Dominance Submission Competition	R&D (dev.) Project marketing R&D (leverage) Commer-cialisation	Project resources, problem-solving skills Organisational resources Individual skills, Reagenix account Knowledge of tool markets, business skills
Non-influential: Funding bodies, Industrial customers		**Reasons**: Funding bodies had no control over Reagenix, they just financed projects where it was applied. Industrial customers had, in practice, no control over the work either.		

The firm K put forward the original vision and had both technical skills and organisational resources for making it possible, but more as a vision of the future of one company than as an innovators' dream of changing the whole paradigm of industrial embedded software engineering. By the time of the Sokrates project proposal, the latter was prevailing. The organisational dominance of the firm K, based on its position as a purchaser, decision maker, and key customer of a red project relationship, had also ended.

The VTT managers and focal researchers co-operated with each other and third parties to launch activities for making the researchers' vision true. The managers used their contacts with Tekes and industry, as well as their project planning skills, to market the Sokrates project. They showed rather little interest in taking part in forming the technical code generation vision. As an example, I myself prepared a separate Finsoft project proposal, based on my own software engineering vision. This proposal was, along with the Sokrates proposal, sent to Tekes. Only the latter was accepted. KTM did not influence the other parties during the Speco period similarly to Tekes, although it did finance the Draco project. The reason for this was KTM not employing any organisational means of controlling the work, such as project management groups. The Draco project was co-ordinated by an internal interest group of one manager and a few researchers.

During the Sokrates period, the focal researchers remained the most influential group of actors, but their form of interaction changed from co-operation to dominance. The reason was that they now had all the project resources needed to realise the code generation vision. The focal researchers controlled closely all the human, technical and physical resources of the project. As a small but illustrative example, the workstation computer that was purchased by TKO for Sokrates was not connected to the computing network used by the rest of the laboratory. Only the project members had the access to the computer. Tekes submitted to the dominance of the researchers. It carried out co-ordination and evaluation work at the Finsoft program level. Industrial interest was enough for Tekes to be assured that the project was useful.

Industrial companies such as Nm co-operated with the focal researchers, for they needed support for exploiting the project results. The needs of the firm Nm were taken care of by TKO, due to the fact that it was necessary for TKO to show the industrial applicability of the research results. The VTT managers, research customers and the Sokrates steering group parties did not control the R&D process, they only co-ordinated it. Colleagues did not even co-ordinate the process, but commented its results. Therefore, these groups of actors were actually non-influential during the Sokrates period. The reason for the perhaps surprising lack of influence of the managers was that the researchers were now resource-wise nearly independent.

The Reagenix period shows a radical change in the forms of interaction of the groups of actors. The VTT managers dominated during the beginning of the period, owing to their ability of controlling the project marketing process and organisational resources. The reason for the rapid change of the focal researchers from the most influential group of actors to an almost powerless group was the lack of financially considerable red or blue code generation projects after Sokrates. Although the focal researchers possessed technical code generation skills, the income from the code generation projects after Sokrates did not cover the cost of their salaries. The Reagenix license fee account would have ensured some kind of financial independence, but the income appeared to be very modest.

Because of the dramatic decrease in financial and other process-related resources, research customers become later on the most important group of actors for the further leverage and exploitation of the code generation competence. They were both the biggest and most faithful group of actors during the middle and late nineties with regard to interest in utilising the code generator. Furthermore, industry was involved in their projects as research partners. Since Tekes had changed its funding policy towards industrial needs, it was common that companies took part in blue research projects as active research partners with concrete problems to solve.

The interaction with tool vendors became competitive after the introduction of the Reagenix code generator. However, since the tool failed to become a truly commercial alternative, e.g. for the code generator of the firm Is, the interaction changed to a don't care situation. The EDA tool business grew in Finland to a recognised sector, where knowledge of the domestic market and software business skills played a central role. The selling of Reagenix licenses did not became any comparable business at all. Although there are no exact figures available, the few licenses sold by VTT were most likely only a tiny fraction of the domestic CASE tool market.

During the Reagenix period, funding bodies and industrial customers remained non-influential groups of actors, although they participated both in the red and blue code generation related projects. These actors could not control the work carried out to apply and develop the Reagenix code generator further. The results of the interview of the firm Nm show that even the most important industrial customers were led by the technology experts, i.e. the focal researchers.

The companies buying tool licenses had even less influence, unless there was a project involved in connection with which the license was purchased. Although funding bodies financed blue research projects in which the generator was applied, they did not have the means of affecting the technology. Not even the technical experts of these projects had such skills, due to the experts being end-users rather than developers of the code generation technology.

7.3.2.2 Mutuality – scanning and diffusion

The change of the focal nets outlined in the previous sections will now be analysed at the level of relationships between groups of actors. Analysis will be carried out by using the functions of assuring and balancing of mutuality based on scanning and diffusion by socialisation or combination. The changes of the process and project nets are viewed from the perspective of the relationships of the focal code generation researchers in Table 18, since the code generation competence to be analysed based on the change of the focal nets was created and possessed by this group of actors within the focal organisation. For the same reason, the final row of the table is devoted to make explicit the change of interrelationships among the focal researchers themselves.

Table 18. Change of the mutuality of the focal researchers.

Actor groups	**Management of changes**			**Mostly affected elements**
	Speco (1985–1987)	Sokrates (1988–1991)	Reagenix (1992–1998)	
VTT managers	Assuring of mutuality (+): Diffusion by combination (project proposals)	Balancing of mutuality (+/x): Diffusion by combination (Sokrates plans and results)	Balancing of mutuality (x/–): Diffusion by combination (code generation tools and methods)	Resource collections (code generation tools, methods), Activity structures: (project tasks)
Industrial customers	Balancing of mutuality (x): Scanning by combination (application understanding)	Balancing of mutuality (x/+): Diffusion by combination (Sokrates plans and results)	Assuring of mutuality (+): Diffusion by combination (code generation tools and methods)	Resource ties, constellations (code generation tools and methods), Activity links, patterns: (project tasks)
Tool vendors	Assuring of mutuality (+): Scanning by combination (related work)	-"-	Balancing of mutuality (x/–): Diffusion by combination (tools)	Resource constellations (code generation tools and methods)
Research customers	–	Assuring of mutuality (+): Diffusion by combination (help to use the generator)	Assuring of mutuality (+/x): Diffusion by combination (help to use the code generator)	Resource constellations (code generation tools and methods)
Funding bodies	Balancing of mutuality (x): Diffusion by combination (Draco proposal)	Assuring of mutuality (+): Diffusion by combination (Sokrates proposal, plans and results)	Balancing of mutuality (x): Diffusion by combination (Kaapeli and MCS-REA proposals)	Activity patterns (project management tasks), Resource constellations (project resources)
Colleagues (internal collabo-ration)	Assuring of mutuality (+): scanning by combination (Draco and Sokrates proposals)	Balancing of mutuality (+/x): diffusion by socialisation (failure to establish the R Group)	Assuring of mutuality (–): (decreasing interest in code generation R&D)	Speco/Sokrates: organisational structures (co-ordination) Reagenix: -- (only informal interaction)
Focal researchers	Assuring of mutuality (+): scanning by combination (Speco and Draco results)	Assuring of mutuality (+): diffusion by socialisation (joint work in the Sokrates project)	Assuring of mutuality (+/x): diffusion by socialisation/ combination (share of the generator tool)	Activity structures (informal and project related) and resource collections (project results)

The key elements of the relationships affected by the changes, be they resource (collections, ties, constellations), activity (structures, links, patterns) or actor (structures, bonds, webs) related, are also identified. Increasing mutuality between the focal researchers and the other groups of actors, as it was identified from the case data and will be explained below, is marked with "+", decreasing mutuality with "–" and the situation where the mutuality did not change with "x".

VTT managers

During the Speco period, the managers co-operated with the focal researchers by assuring mutuality based on diffusion by combination. The purpose was to produce proposals and plans together for launching code generation related projects, by referring to and integrating explicit pieces of knowledge. During the Sokrates period mutuality clearly decreased. The two groups still handled project plans and non-technical reports together, but now the managers together with the steering group occasionally expressed their concern about the way the information was delivered.

During the Reagenix period mutuality further decreased, although some planning and co-ordination of the exploitation of the code generator was carried out jointly. Resource collections became more important than activity structures between the two groups of actors. Especially during Sokrates, joint project planning and management activities had been the key elements in their relationship. Later, when joint project preparation activities decreased, the groups were related to each other mainly via the collection of technical code generation resources. The researchers developed and marketed the resources to a large extent by themselves, the managers were involved, for example, in signing license agreements on the behalf of VTT and in other similar administrative tasks – not in co-ordinating the actual R&D process as during Speco and Sokrates.

Customers

Resource constellations and ties were also the dominating elements in the relationships between the focal researchers and industrial customers, exceptions being perhaps such red projects as MCS-REA, where the activities of the customer and the researchers were closely linked together. During the Speco period the researchers were scanning explicit knowledge to understand the product application of the firm K, but the balancing of mutuality of the project activities failed and a dispute resulted. During the Sokrates and Reagenix periods mutuality between the two groups increased. It was then based on diffusing explicit code generation research results to industrial use. However, the designers of the firm K, for example, resisted the pilot use of the code generator vigorously (cf. Appendix 2, page 6).

Regarding relationships with tool vendors as special kinds of industrial customers, instead of increasing the mutuality of activities decreased during the Reagenix period. The relationship between the firm Is and the researchers became competitive.

For this reason, it was no longer possible to carry out joint activities. In more general terms, tool vendors were interested in the results of the research, i.e. in resource ties and constellations. Activity links and webs remained weak: the firm Is took part in the steering group of Sokrates, and some studies were carried out at TKO based on discussions with another vendor. However, the two groups never linked their activities together as tightly as, for example, the firm Nm as an industrial customer and the focal researchers in the Kaapeli and MCS-REA projects.

The relationships between the focal researchers and their research customers revolved also around resources, especially the Sokrates and Reagenix generators, because the latter group carried out research on other subjects than code generation. The activity links and patterns created involved assistance in the use of code generation techniques and tools. The mutuality of such activities was rather high, although the focal researchers were not notably involved in the external patterns of activities and webs of actors of the blue projects in which the Reagenix code generator was used - e.g. in the R&D tasks the members of the projects carried out together with the industrial members of the steering groups of the projects or with other research institutes.

As stated by the researchers, they also became tired of the fact that hardly any financial or other types of rewards were given to them despite of their interest and assistance in the use of the generators in blue research projects. Furthermore, their assistance was not always desired, even if they were willing to offer their help. Earlier, when some of the research customers were the former Sokrates project team members, e.g. in the Tulko and Diag projects (Table 16), help was asked for and provided regularly.

Funding bodies

During the Speco project the mutuality of activities between KTM and the focal researchers was very low. During the Sokrates period mutuality increased considerably. Tekes funded Sokrates and its early exploitation projects. However, mutuality decreased very rapidly after the Sokrates project had been finished. The preparation of the MCS-REA project proposal, which included the acquisition of funding from Tekes to the firm Nm, was the last considerable joint code generation related activity pattern in which both Tekes and the focal researchers were involved.

Colleagues

During the Speco period mutuality was high between the focal researchers and their colleagues – who basically included all other software engineering researchers of the focal organisation. They carried out joint planning and co-ordination activities to scan information on existing approaches and to plan for the Sokrates project. During the Sokrates period there was an attempt to continue increasing mutuality, in the form of an interest group called the “R Group” that would have helped diffusion by socialisation.

This group was established to carry out joint planning and co-ordination of R&D on so called concurrent systems, for which the code generation techniques were also targeted. However, the attempt failed and the interest group was dissolved as an organisational structure. Mutuality further decreased during the Reagenix period, because the colleagues were not involved in code generation related matters at all.

Researchers themselves

The researchers kept the mutuality of their joint activities high trough all three periods, except perhaps the very late nineties, when many of them had already left VTT. During the Sokrates project implicit knowledge was effectively diffused within the project group, which looked like a very homogenous team to the outsiders. Earlier, the explicit results of the Speco and Draco projects had been shared to prepare and launch the Sokrates project. Line organisation structures, such as the focal Software engineering research group, did not play any considerable role compared with project-related activity structures.

Most of the shared resources were technical results and certain kinds of problem solving skills. Later, when the Sokrates project team had been dissolved, the diffusion based on explicit resources gained prestige, due to the fact that joint project activities were decreasing considerably. Many of the researchers left VTT and certain differences in viewpoints seem to have emerged: "From my point of view, it was a mistake that it [the Reagenix code generator] was taken as the continuation of Sokrates and therefore no one was interested".

To summarise the developments with regard to the mutuality of the activities of the focal code generation researchers, mutuality towards:

- the VTT managers decreased (only a few joint activities),
- industrial customers was high (activity links, resource ties), except with tool vendors due to competition,
- research customers decreased (no joint R&D activities),
- Tekes decreased (no funding for code generation research),
- colleagues decreased (only informal interaction), and
- mutuality between the focal researchers themselves remained high, except in the late nineties (only a few persons stayed at VTT).

I will now continue the analysis of the code generation case by investigating the competence marketing and purchasing, R&D, competence exploitation, and co-ordination processes associated with the project nets.

7.4 CONTENT OF LOGIC OF ACTION – CODE GENERATION PROCESSES

Table 19 provides an overview of code generation related processes carried out from 1985 to 1998. Key activities performed in these processes constitute the implementation of the logic of action of the groups of actors involved in the nets. In the following, I will discuss the evolution of each of the processes shown in Table 19 during the three successive periods.

Table 19. Overview of code generation related processes.

Processes	**Main types of activities**			**Project nets**
	Speco 1985–1987	Sokrates 1988–1991	Reagenix 1992–1998	
Marketing (VTT managers) *Purchasing* (industrial customers)	Initiating of Speco and Draco Planning of Sokrates --	Marketing and initiating of spin-offs Use of Sokrates Planning of exploitation projects	Marketing and initiating the use of Reagenix Purchasing of IPRs and tool licenses	Speco: Tekes, KTM, industry Sokrates: VTT, Tekes, industry, tool vendors Reagenix: VTT, educational institutes, industry, IPR buyer company
R&D (code generation resear-chers)	Evaluation of Draco, Refine Development/ evaluation of a prototype of a code generator	Development of the Sokrates method, tools, toy elevator, system solutions Writing of courseware and manuals	Development/ evaluation of Reagenix, Reanimator, Cute and ReagOS Writing of courseware and manuals	Speco, Speco-2, Draco, Sokrates, Reagenix license fee account, Aniprosa, Aniprosa-2 License selling and training relationships
Exploi-tation (customers)	Demonstration of the results of Speco Joining of the Sokrates steering group	Use of the Sokrates results Use of technical documents Taking part in SA/SD courses	Conducting of red spin-off projects Exchange of licenses Acquisition of a patent Use in blue projects Taking part in courses	Synchro, Sasic, Tulko, Kaapeli Turva, Diagnostics MCS-REA, VTT Electr. Laboratory Osdyn, Raski, Kaasu, Sympa, Cute, Nosto, Nosto-2, Rulla, Rulla-2, Rekki, AVV, Mots-2
Control & co-ordi-nation (VTT managers, Tekes)	Control of the Speco and Draco projects	Control of the Sokrates project and the Finsoft program Planning for spin-offs	(Planning for continuing projects) Control of spin-off projects	Project steering and management groups, TKO research council, schaffners, TKO management group, R Group

Marketing vs. purchasing

The focus of the *marketing and purchasing processes* was on blue and red projects during the Speco and Sokrates periods. The corresponding activities involved the kinds of project marketing meetings and preparation of project proposals and plans illustrated in [Seppänen et al. 1999a]. To put it simply, the focal researchers and managers were the marketers and Tekes and industrial parties of the Finsoft research program the purchasers.

The marketing process was owned by the VTT managers, though, because they were controlling the organisational resources needed for planning, carrying out and co-ordinating the process as a whole. For example, a frame of hours was fixed for marketing in the annual plans of the organisational groups. The hours actually spent on marketing were recorded to a certain activity code that was controlled by group managers. As the code generation stories in Appendix 2 indicate, the focal researchers felt that they had not enough control over marketing activities: “An ordinary researcher is not allowed to talk to industrial managers. The section head on his own makes the decisions on the use of the section’s resources". The managers felt that the researchers were not active enough in launching new projects.

During the Reagenix period, the focal researchers themselves carried out most of the activities related to selling of tool licenses, without much co-ordination with the managers who prepared the operational and marketing plans. The code generator was actively marketed by the VTT managers from about 1992 to 1994, especially when one of the focal researchers deputised me as a section head. Later in the nineties, almost the only considerable code generation marketing activity of the managers was to sell the code generation rights to the company established by the former researchers. Some inquiries concerning the rights had been made earlier by other parties, but the researchers had turned them down.

Marketing to research customers was mentioned in, for example, plans of the focal Software engineering research group. However, it was impossible to recognise any planned marketing activities from the case data directed to research customers. Moreover, some customerships were created through former Sokrates project team members working in new projects. Although research customers, in the end, became the most influential group of actors in the nineties, the true owners of the competence purchasing process in the business sense were the industrial customers.

Since research customers did not provide income for the focal organisation, the VTT managers who owned the marketing process did not actively promote the use of the generator in blue research projects. The same holds for tool licenses given to educational institutes in connection with courses that the researchers gave on the design method. This was not viewed as business by the managers.

Research and development vs. exploitation

The focal researchers were the owners of the *R&D process* throughout the three periods, because the skills to carry out research on code generation were neither diffused to customers nor to VTT colleagues. Rather than a linear sequence of activities from innovation to commercialisation, this process involved several intertwined R&D cycles. As exploiters of the results produced by the focal researchers, customers performed technology screening, evaluation and utilisation. However, the customers did not build any significant code generation competence of their own.

Some customers, such as the firm N, exploiting one of the developed system solutions, were capable of building competence on their own. Many others, such as the firm Nm, did not have the necessary capabilities, although this was expected by the researchers. Selling of licenses was a new type of activity in the focal organisation, as was actually also the writing of professional-style technical users manuals and course material on code generation. These activities can be considered as the "commercialisation" phase of the R&D process.

The rather loose integration of the R&D process with the customers' *competence exploitation process* can be seen as part of the relationship strategy of the focal researchers: make it fast and cheap and deliver value, not work to customers. One of the results of this strategy was that there were only few, if any, relationships in which external parties would have been competent enough to develop code generation skills themselves. The focal researchers themselves stated that their purpose was to transfer the skills "of using technology" to customers, not the skills of developing the technology. This is the considerable difference between the code generation and the fault diagnosis cases.

Research customers can be considered having owned the exploitation process, taking into account the role of the Sokrates steering group members in the early phases of exploitation and the number of blue projects in which the generator was used in the nineties. Although the results of the Speco project remained just a demonstration system, they were paving the way for the Sokrates project. Another demonstration system was also built in connection with Sokrates, but for most members of the steering group, technical documents remained the only tangible results of Sokrates.

However, the research and industrial customers exploited the code generation competence elements in a rather versatile manner. The Reanimator graphical debugger was perhaps the only element the usefulness of which was suspected even by the focal researchers. They emphasised the profitability of courses given to students and industrial professionals, although the courses did not belong to the core activities of VTT.

Co-operation

The *controlling and co-ordination process* was owned by the VTT managers. Tekes controlled, in principle, the R&D process during Sokrates, but only indirectly through funding and the Finsoft program level activities.

To summarise, the focus of the process portfolio was twofold *during the Speco period*, project marketing and purchasing on one hand, and early innovative R&D on the other – the vision was thus marketed and created at the same time. These processes progressed in parallel and in a reversed mode compared with the ideal project portfolio from green to red projects. One of the reasons for this was that a few key people at VTT and at the firm K shared a vision of automatic software production and could therefore give a jump start to the work. Both parties aimed at producing the first tangible results soon and using them for launching more comprehensive development activities. Less emphasis was put on the other processes than the actual research and development process. For example, the control and co-ordination of the Draco project was carried out mostly on an informal basis. From the viewpoint of the process portfolio, the period could be characterised as "Testing of an industrial vision".

During the Sokrates period, the focus was also twofold, but now on the R&D and competence exploitation processes. Project marketing and purchasing were carried out on the side of the Sokrates project, involving mainly the companies participating in its steering group. The Electronics laboratory of VTT that became interested in the results was located on the same premises as TKO and even had its people in the steering group of Sokrates at some point. No great marketing and purchasing efforts were thus needed for initiating the early exploitation activities. Writing manuals on the use of the code generation technology and giving courses on the Sokrates-SA design method may have been more oriented towards marketing, in addition to the ultimately failed discussions with the firm Is. Although the whole Finsoft program was co-ordinated in a very structured manner, the focal researchers kept a tight control over research and exploitation, due to their technical expertise. The period could be characterised as "Demonstration of a research vision". The limits of technology that Tekes wished to be explored were not in sight yet, since the major example of the Sokrates project was only a toy system.

During the Reagenix period, everything changed in the process portfolio. The large-scale R&D activities carried out in Sokrates finished almost entirely. The Reagenix code generator was a kind of garage product compared with the Sokrates results in which three years and notable sums of money were spent on. The competence marketing and purchasing processes became critical for the continuation of the R&D process. The exploitation of the generator was quite sporadic, only relatively small activities were carried out for individual industrial and research customers. In research projects the exploitation was controlled by research customers, while the code generation researchers did not play any central role. In industrial projects, the situation was slightly different, but it was more important to conduct R&D than to plan for and control competence marketing. Also the organisational control of the R&D process corrupted rather rapidly due to the managers losing interest in code generation as a business. The informal co-ordination process between the focal researchers faded away slower, as a result of people leaving VTT. From the viewpoint of the process portfolio, the Reagenix period could be characterised as "Fighting for the vision".

7.4.1 Activities on resources – assuring of capability

Table 20 summarises changes in the code generation competence during the three periods, using the framework shown in Figure 17. Assuring of capability involves the management of the life-cycle of resources, on which competence is based. The life cycle of product-related resources consists of the phases of incremental changes, discontinuity, substitution, competition and dominance. They include design and solution elements of human, technical and physical resources, and their content is viewed from the four dimensions of applications, functions, techniques and technologies. Process-related resources include temporal and financial resources, and reputation is an important organisation-related resource.

As pointed out in Chapter 4, the main purpose of capability assurance is to manage inconsistencies in relationships. In practice, this can be performed not only as part of the competence marketing and purchasing processes, but also in connection with the control and co-ordination of ongoing R&D and competence exploitation processes. In Table 20, management of inconsistency is identified, based on the case data, as being good (+), poor (–) or moderate (+/–) for each type of code generation related resources from 1985 to 1998. Justifications of these valuations will be discussed below.

Speco was a period of discontinuity in terms of the life cycle of code generation related product resources: code generation would mean radical changes in software development as industrial work and code generators were new as a tool technology. Contrary to the scientific and early commercial approaches to code generation, the focal researchers chose to use the SA/SD system design technique as their starting point. This technique dominated industrial embedded systems design in the late eighties, and some of the focal researches had "fallen in love" with the technique. It was not a bad choice, concerning the issue of helping the researchers and industrial software engineers to understand each other.

During the Sokrates period, code generation design and solution elements were developed to substitute existing elements, except for the Sokrates-SA design technique, which was an incremental extension of the SA/SD technique. The solution elements were generic technological implementations. It was not necessary to have any detailed knowledge of the functions they implemented and the techniques upon which they were based, to be able to exploit them. This was also the goal of using the code generator, but its achievement could not yet be shown: other VTT researchers were the main group of users.

The life cycle of product-related resources changed again *during the Reagenix period*. Only incremental changes were required to improve the code generation functions. The dominance of the SA/SD design technique was weakening due to object-oriented techniques, competition emerged regarding commercial code generators, and in most applications where code generation was used, changes remained incremental rather than radical. This was the case also in the firm Nm, which was a key customer. Technological solution elements remained in their substitutive phase.

Table 20. Assuring of code generation capability by VTT.

Speco (1985–1987)	**Sokrates (1988–1991)**	**Reagenix (1992–1998)**
Product: human, technical and physical resources		
Design elements: – functions: discontinuity (innovation of code generation) – techniques: dominance (SA/SD widely used) – technologies: discontinuity (use of code generators) – applications: discontinuity (change the way to develop software) Solution elements: --	Design elements: – functions: substitution (use of code generation instead of manual coding) – techniques: incremental changes (Sokrates-SA) – technologies: substitution (wide use of generators) – applications: discontinuity (change the way to develop embedded software) Solution elements: – technologies: substitution (replace existing solutions)	Design elements: – functions: incremental changes (improvement of code generation) – techniques: substitution (SA/SD dominance is over) – technologies: competition (commercial tools exist) – applications: incremental changes (many different applications addressed), Solution elements: – technologies: substitution (replace existing solutions)
Management of inconsistency of product resources between 1985–1998 Design elements: – applications (+/–): trying to cause a radical change in the industrial development of embedded software, ending up to deal with many kinds of applications – functions (+): R&D and gradual improvement of code generation principles – techniques (–): improvement of SA/SD, while competing techniques emerged – technologies (–): trying to commercialise the code generator in a competing situation Solution elements: – technologies (+/–): attempting to replace existing solutions, partially succeeding		
Process: temporal and financial resources		
Draco: resources gained and managed as planned Speco: disagreement over temporal resources caused problems with firm K, financial resources available to make the initial innovation	Sokrates: success in gaining financial resources was better than planned, problems with temporal resources Other projects: financial resources moderate, temporal resources good	Green projects: very modest financial resources Blue projects: no income from temporal resources Red projects and licenses: disagreement over financial resources with managers, Few temporal resources
Management of inconsistency of process resources between 1985–1998 – temporal resources (–): management of resources by focusing on the analysis of problems and then trying to solve them quickly caused problems in Speco, resulted in unfinished work in Sokrates and was criticised by the VTT managers during Reagenix – financial resources (+/–): financial resources were good during Speco and excellent during Sokrates, but then rapidly collapsed due to the lack of red projects		
Organisation: reputation and other organisational resources		
Reputation: leading embedded software engineering research unit Other: strong role in national R&D initiatives	Reputation: few problems with the steering group and with some colleagues Other: investments in equipment, co-ordination	Reputation: conflict with the firm Is, dispute with the VTT managers Other: few resources provided by VTT
Management of inconsistency of organisational resources between 1985–1998 – reputation (+/–): deterioration of the good reputation among the managers; rather good reputation sustained among the early and late internal and external exploiters – other resources (+/–): organisational support during Speco and Sokrates changed to a doubtful situation during Reagenix, resulting in the loss of organisational visibility.		

The capability of code generation related *product resources* is decreasing during the three periods, except code generation functions that were gradually improved from their early innovative stage. Enhancing the capability of the technological design resources, most notably the Sokrates and Reagenix code generators, did not succeed. The kind of commercialisation at which the researchers aimed also failed. Furthermore, there were problems in the capability of the SA/SD system design technique, which lost its dominating position by the mid-nineties.

The code generation capability concerning automation applications was high in the early nineties, not only among the focal researchers but at TKO as a whole. However, the automation sector recovered very slowly from the recession and the consumer electronics sector for which some code generation projects were also carried out almost disappeared from Finland. No special capabilities were built for the rapidly increasing telecommunication sector. Assuring of capability in terms of the life-cycle of applications was apparently not successful as a whole, the result was a mixed bag of applications with no special attention paid to the enormously growing telecommunication sector. The developed technological solution elements were intended for replacing existing solutions; they succeeded at least partially. However, no considerable business could be created based on these resource elements.

The life-cycle of the *process-related temporal and financial resources* depended much on the phases of the R&D and competence exploitation processes. As an example, financial resources were sufficient during Speco and even better than expected during Sokrates, but they mostly collapsed by 1994. There were problems regarding scheduling and manpower in the Sokrates project, because some of the central tasks could not be finished. In the Speco project, this had caused a dispute with the customer. During Reagenix, resources for carrying out further research and development were very limited indeed. The managers, who controlled the time and money spend on marketing, did not want them to be invested in code generation. In terms of the management of inconsistency concerning process-related resources, the focal organisation thus succeeded mainly in assuring of financial capability during the Sokrates period, which was the competence development and early exploitation phase. This could not assure the continuation of sufficient amount of financing after the Sokrates project.

Capability of *organisation-related resources* was good during Speco and Sokrates, when TKO was centrally involved in code generation related national research and development activities. Later, when the logic of action of the managers and researchers diverged, only few organisational resources were available to the researchers. Disagreements with the firm Is did not seriously affect the reputation of VTT, although they were noticed by some customer companies. The decrease of organisational resources resulted in a rapid loss of the visibility of code generation related activities. What used to be a promising idea during Speco and flagship research during Sokrates, became an almost invisible hobby of a few researchers.

7.4.2 Activities on resources – balancing of particularity

Balancing of the particularity of resources in certain relationships means absorption and problem solving that affect the codification of product-related resources and the contextuality of process and organisation-related resources. Table 21 provides an overview of the balancing of particularity of code generation resources from 1985 to 1998, by means of problem solving (indicated by "–" for decreasing particularity, to create generic and context-independent resources) and absorption (indicated by "+" for increasing particularity, to create tacit and context-specific resources).

Product-related resources

The product-related code generation design elements were quite tacit *during the Speco period*, although the innovative skills of the focal researchers were used for producing the first explicit technical and physical resources. The code generation functions, the prototype of a generator that was built, and knowledge of the applications of the firm K were all particular to the Speco project. However, the idea of using the generic SA/SD technique that was explicitly described in textbooks and taught in courses was already emerging. As an alternative, it would have been possible to use application-specific design techniques and languages, as in the Draco tool evaluated in the Draco project. This would have meant high particularity of the developed design elements also with regard to the technique dimension.

During the Sokrates period, the code generation functions became portfolio-specific, since they produced software based on a certain program transformation logic. The Sokrates-SA design technique, which was in the heart of this logic, can thus also be considered as portfolio-specific. Even the Sokrates generator was not fully generic, but designed to support code generation from structured system models, especially from the so-called state transition and functional models. When the customer focus was directed to machine automation, it can be said that the particularity of the product-related design elements as a whole turned into portfolio-specific.

The solution elements developed in Sokrates were generic, and remained as such also *during the Reagenix period*. The Reagenix code generation functions and generator were also generic, i.e. the particularity of the code generation functions and their implementation technology decreased. The Sokrates-SA based design technique was replaced by a generic component-based approach that could be tailored for specific needs. A generic visual modelling tool was used for defining and modifying the notation. This relieved the researchers of the need to use the CASE tool of the firm Is as a part of their tool environment. Since there was no focus on specific applications any more during the Reagenix period, the overall particularity of product-related resources decreased, all of the resources were quite generic. The continuous decreasing of the particularity of the resources fits with the logic of the focal researchers in creating a generic, well-packaged code generation solution. The decreasing can also be seen as a result of the change of focus on external activity links and patters and resource ties during Speco and Sokrates to internal resource collections during Reagenix.

Table 21. Balancing of particularity of code generation resources by VTT.

Speco (1985–1987)	**Sokrates (1988–1991)**	**Reagenix (1992–1998)**
Product: human, technological and physical resources		
Design elements: – functions: tacit (specific code generation rules invented for the firm K) – techniques: generic (Ward-Mellor SA/SD) – technologies: tacit (code generator prototype) – applications: tacit (individual skills in the application of the firm K) Solution elements: --	Design elements: – functions: portfolio-specific (code generation applied in several projects) – techniques: portfolio-specific (Sokrates-SA) – technologies: portfolio-specific (SA/SD generator), – applications: portfolio-specific (focus on machine automation applications) Solution elements: – technologies: generic (widely usable solutions)	Design elements: – functions: generic (code generation applicable to many different problems) – techniques: generic (Reagenix models) – technologies: generic (Reagenix code generator), – applications: generic (no specific focus on certain applications or problems), Solution elements: – technologies: generic (widely usable solutions)
Absorption (+) and problem solving (–) of product resources in 1985–1998 Design elements: – applications (+/–/–): starting from a specific product application, trying to focus on machine automation and ending up with dealing with various types of applications – functions (+/–/–): from specific via SA/SD based to generic code generation rules – techniques (+/–/–): changing of SD/SD to Sokrates-SA and then to Reagenix models – technologies (+/–/–): development of a commercial generic code generator Solution elements: – technologies (x/–/–): development and utilisation of generic solution elements		
Process: temporal and financial resources		
Draco: generic (loosely controlled evaluation of new technologies) Speco: portfolio-specific (resources provided as part of strategic co-operation between TKO and the firm K)	Sokrates: portfolio-specific (resources provided as a part of the Finsoft program) Other projects: context-specific (participation as experts on an individual basis and occasionally)	Green projects: context-specific (occasional) Blue projects: context-specific (occasional) Red projects/licenses: portfolio-specific/generic (resources provided within projects and from licenses)
Absorption (+) and problem solving (-) of process resources in 1985–1998 – temporal/financial resources (–/–/+): most of the resources were portfolio-specific during Speco and Sokrates, during Reagenix most resources were context-specific		
Organisation: reputation and other organisational resources		
Reputation and other resources: portfolio-specific (contractual project-based embedded systems R&D in Finland)	Reputation and other resources: portfolio-specific (no change to the earlier situation)	Reputation: context-specific (organisational restructuring resulted in a greater heterogeneous unit) Other: context-specific (project-based resources)
Absorption (+) and problem solving (–) of organisational resources in 1985–1998 – reputation (–/–/+): TKO created a portfolio-specific reputation based on embedded systems research and development, ELE needed to renew the reputation in the nineties – other resources (–/–/+): project-based planning and management of R&D activities increased uniformity of the institute; this caused problems in license selling and small-scale code generation problem solving tasks		

Process-related resources

On the basis of the case data, particularity of process-related resources does not directly correspond to particularity of product-related resources, except during Sokrates when most temporal and financial code generation resources were portfolio-specific. The resources had been portfolio-specific even in the Speco project, which was a part of the strategic alliance between the firm K and TKO. The particularity of process-related resources became high at the end of the Sokrates period and especially afterwards. The resources depended on projects in which the focal researchers happened to participate and the licenses they were every now and then able to sell.

Therefore, while the particularity of product-related code generation resources gradually decreased and they became very generic, the particularity of process-related resources increased rapidly, partly due to changes in project nets. The result was that highly generic technical and physical resources were offered in connection with highly specific financial and temporal resources. The comparison of the Kaapeli (1990–1991) and the AVV (1995–1997) projects illustrates the change that took place. The former was carried out for the firm Nm, which wished code generation functions to be tailored for a specific programming language, with an intention to use them manually. The project was one of the first spin-offs of the Sokrates project. The code generation rules and their use were highly particular to the needs of the customer, whereas the work was carried out as a part of Sokrates and its early exploitation projects.

None of the focal researchers took part in the AVV project, where the Reagenix code generator was used as an off-the-shelf tool by three software safety researchers of VTT, to produce a certain simulation program. The schedule, volume, budget, tasks and human resources of the AVV project determined fully why and how the generator was used. The focal researchers offered to help in the use of the generator, but "no help was needed". They did not receive any income from the project either. A fully generic code generation solution was thus used in a one-of-a-kind project.

Organisation-related resources

Particularity of the organisation-related resources follows a pattern similar to process-related resources. In the late eighties and early nineties TKO aimed at creating a reputation of an embedded systems expert organisation, whose main products were demanding research and development projects. In other words, TKO wanted to make use of portfolio-specific project-based organisational resources for the needs of industrial customers interested in developing or applying embedded systems in their products. The organisational uniformity of project-based R&D activities have become even more important since 1994, when ELE, a larger and more heterogeneous unit compared to TKO, was formed. The particularity of the idea of selling licenses and the focus on internal use of self-developed design techniques and tools were relatively high, compared with the aim of increased organisational uniformity at ELE.

7.5 EVOLUTION AND VALUATION OF THE COMPETENCE

I will now discuss the evolution of the code generation competence at VTT from 1985 to 1998. The integration of competence elements will also be addressed, as well as my interpretation to the valuation of competence elements by different parties. The integration of competence elements may involve resource collections and activity structures (in the focal organisation), resource ties and activity links (in relationships), or resource constellations and activity patterns (in networks). Competence elements may be strengthening, competing, complementary or neutral with regard to integration with other elements.

The longitudinal value of competence is viewed from the perspectives of one supplier and several purchasers, concerning the expected, perceived and historical values to the extent that they could be identified from the case data. Expected value refers to the valuation of competence before its development or exploitation, i.e. it is a kind of *market value* of the competence. The perceived or *delivered value* of competence is evaluated during and at the end of its development or exploitation. Expected and perceived values of competence were analysed in this research mainly on the basis of documentary data. Historical or *usage value* refers to the valuation of competence after its development or leverage, analysed in this research on the basis of interviews.

7.5.1 Competence during the Speco period

Table 22 summarises the competence elements possessed or created by VTT during the Speco period, as well as their planned and realised integration and their valuation. *Product-related competence* elements involve skills of developing embedded computer system technologies, knowledge of their design techniques, skills of evaluating and developing new technologies and capabilities in machine automation applications. The latter were in part absorbed from the firm K, which possessed also the other competence elements. The firm K wished to lead the Speco project also in technical terms, which created competition on how innovative research should be carried out. However, most of the elements strengthened both internal activity structures and resource collections of TKO and the activity links and resources ties with the firm K. The capability of TKO to understand automation applications complemented its resource ties with the firm K.

The focal researchers considered the historical value of all of the product-related competence elements as very high (++) or high (+), whereas the VTT managers were somewhat more critical in their valuation. They considered the historical value of tool development skills as low (–), but the value of knowledge of system design techniques was considered high also by them. Tekes valued product-related code generation competence during the Speco period as high, but it was not yet directly involved in code generation related activities. The firm K considered the historical value of the elements as moderate (+/–) or low, especially the skills of TKO in developing new design tools.

Table 22. Code generation competence during 1985–1987.

Competence elements: Owners	Planned/realised integration of elements	Supplier valuation	Purchaser valuation
		(E)xpected, (P)erceived, (H)istorical	
Product-related competence elements			
Skills to develop embedded computer system technologies: TKO (possessed also by the firm K)	Strengthening of internal resource collections/ok Strengthening of resource ties with the firm K/ok	Researchers: H ++ Managers: E ++, P +, H +/–	Firm K: E ++, P +, H +/– Tekes: H ++
Capabilities in machine automation applications: TKO (absorbed in part e.g. from the firm K)	Complementing of resource ties with the firm K/ok	Researchers: H + Managers: E ++, P +, H +/–	Firm K: H +/–
Knowledge of embedded system design techniques (especially SA/D) : TKO (possessed also e.g. by the firm K)	Strengthening of internal resource collections/ok Strengthening of resource ties with the firm K/ok	Researchers: H ++ Managers: E ++, P +, H +	Firm K: E ++, P +, H +/– Tekes: H +
Skills to forecast, evaluate and develop new embedded system design tools: TKO (possessed also e.g. by the firm K)	Strengthening of internal activity structures/ok Strengthening of activity links with the firm K/competing	Researchers: H ++ Managers: E ++, P –, H – Colleagues: H +	Firm K: E +, P +/–, H – Tekes: H +
Process-related competence elements			
Skills in (team-based) solving of complex computer-related problems: focal researchers	Strengthening of internal activity structures/ok Strengthening of activity links with the firm K/competing	Researchers: H ++ Managers: H + Colleagues: H +/–	Firm K: E ++, P –, H –
Skills to initiate and conduct embedded software engineering R&D: TKO (especially managers and senior researchers)	Strengthening of internal activity structures/ok Strengthening of activity links with industry/ok Strengthening of actor bonds (and activity links) with Tekes/ok	Researchers: H ++ Managers: H + Colleagues: H +/–	Tekes: P ++, H ++ Industrial parties: H +
Organisational competence elements			
Skills to establish and use contacts with Tekes and industry: TKO	Strengthening of actor bonds and webs with external parties/ok	Researchers and managers: P: ++, H ++	Tekes: H ++ Industry: H ++
Parties isolated from the competence elements: KTM: provided funding for the Draco project, but was not otherwise involved			

The two most important elements of *process-related competence* were team-based problem solving knowledge of the focal researchers and the organisational skills of launching R&D activities. These elements strengthened internal activity structures, external activity links with industry and external actor bonds especially with Tekes. An exception was the dispute with the firm K on how to carry out the innovative research process. However, this did not break the mutual actor bond or even the activity link, although the firm K perceived it as an indication of low competence.

Some interviewed colleagues criticised the organisational processes used for controlling the embedded software engineering research at TKO, but all the other parties evaluated the capability of TKO in this regard as high or very high. The fact that TKO was a key player in Finsoft, owing to its remarkable involvement in conducting and co-ordinating applied engineering research, indicates the same. This capability was provided together with knowledge of embedded systems design techniques, which was clearly valued as the best element of product-related code generation competence of TKO during the Speco period. Together with the main *organisational competence* element, the skills of establishing and making use of contacts with both Tekes and industry, they were paving a way for a pragmatic, yet innovative approach to code generation. The organisational skills of networking strengthened actor bonds and webs. Considerable activity links existed only with the firm K, but the skills of establishing contacts helped to create new activity patterns related to Sokrates.

KTM was isolated from the competence of VTT, because it provided funding for the Draco project but was not involved in any other way in code generation related activities.

7.5.2 Competence during the Sokrates period

During the Sokrates period, the versatility of competence elements increased remarkably, as indicated in Tables 23, 24 and 25. Many *product-related competence* elements became portfolio-specific. Functional skills were made explicit via code generation rules. These skills were intended for complementing resource ties with industry, in which knowledge of how to generate computer programs from design models was still rare. In reality, most of the members of the Sokrates steering group did not need their competence to be complemented with these skills, and the firm Is came up with such skills without VTT. The skills in system design techniques turned into portfolio-specific system modelling languages and tool technologies. This provided better means of strengthening resource collections and ties than knowledge of the generic SA/SD method. The skills of developing embedded systems implementation technologies specialised in skills in system solutions that were widely exploitable. Application knowledge became also portfolio-specific, because there was no specific focus on any single application. New application knowledge needed to be absorbed from customers and the Electronics laboratory of VTT, to complement the skills of the focal researchers. However, application knowledge did not play any central role during Sokrates.

Table 23. Product-related code generation competence during 1988–1991.

Competence elements/ Owners	Planned/realised integration of elements	Supplier valuation	Purchaser valuation
		(E)xpected, (P)erceived, (H)istorical	
Skills of developing embedded computer system technologies (system solutions): focal researchers (in part absorbed from the VTT Electronics laboratory)	Strengthening of resource ties (and constellations) with the steering group of Sokrates and with the industrial and research customers/ ok	Researchers: H ++ Managers: P +, H +/–	Steering group: P +, H +/– Tekes: H ++ Evaluators: P +/– Research customers: P ++, H + Industrial customers: P ++, H ++
Knowledge of embedded systems applications: TKO (in part absorbed from customers)	Complementing of resource ties with the industrial (and research) customers/ mostly neutral	Researchers: H ++ Managers: P +, H +/– Colleagues: H+	Steering group: P +/–, H +/– Tekes: H + Evaluators: P + Industrial customers: P ++, H +
Skills of developing and transferring in use design languages, methods and tools: TKO (possessed also by some customers and the VTT Electronics Laboratory)	Strengthening of internal resource collections/ok Strengthening of resource ties with the research (and industrial) customers/ok	Researchers: H ++ Managers: E ++, P +/–, H +/– Colleagues: P+/–, H –	Steering group: E ++, P +, H +/– Tekes: H + Evaluators: P +/– Industrial and research customers: P +, H +/–
Skills of developing manual coding rules and automated code generation functions: focal researchers (also possessed by the firms K, S, N and I)	Complementing of resource ties with the steering group /mostly neutral, in part ok, in part competing Strengthening of resource ties with the customers/mostly ok, in part neutral and competing	Researchers: H ++ Managers: E ++, P +, H + Colleagues: P +/–, H –	Steering group: E ++, P +, H +/– Tool vendors: P +, H – Tekes: H + Evaluators: P +/– Industrial and research customers: P +, H +/–
Parties isolated from the competence elements: Colleagues (especially the ones not involved in the activities of the R Group) (Members of the steering group: mostly did not directly exploit the results) (VTT managers, Tekes: only co-ordinated the R&D process)			

Technological skills of the focal researchers were valued as high by many parties, but the VTT managers considered their historical value as modest, and some evaluators criticised their inclusion in the project. The main competence of the focal researchers was seen in their functional code generation skills. The expected and perceived value of the skills was high, except for the opinions of some colleagues and evaluators.

However, their historical purchaser value was quite moderate among the Sokrates steering group and the early exploiters. The firm Is did not value the skills high enough to exploit them in its own code generator. Skills of developing system design languages, tools and methods can be considered as part of the competence of TKO, not only of the focal researchers. They were valued higher by the researchers than the managers, although some colleagues considered the skills even as a kind of alchemy, or at least not good enough scientifically: "finding the highest point of a fence and going under it". Their historical value was rather low also among many of the steering group members.

According to Tekes, it paid off creating and exploiting the skills, and the turn of the nineties was the right time for doing it. Customers had somewhat varying opinions regarding the historical value of the results in this regard, while only few of them seized the opportunity of making use of them. Knowledge of embedded systems applications was also part of the competence of TKO, but the managers valued its use in code generation research as rather moderate in interviews. The applications in which the results were exploited were diversified. The toy elevator built for demonstrating the results was seen as a good choice by the focal researchers and their colleagues (Figure 21), but some of the steering group members and the VTT managers considered it as too trivial.

In summary, one of the main elements of product-related code generation competence during the Speco period, knowledge of embedded systems design techniques and SA/SD in particular, was reshaped to portfolio-specific skills in languages, methods and tools, while its dependence on certain kinds of applications weakened. The skills could be associated with knowledge of functions for the purpose of generating code from system models. The Sokrates-SA design method, which was described in technical documents, tied the elements together. Code generation functions were the most largely exploited part of the competence, as they were used in the Kaapeli, Synchro and Sasic projects. The tool technologies were still immature. In the Kaapeli project, they were not used at all. The Sokrates-SA design technique included elements, such as an Ada-like design language, the value of which was rated equally low by most of the parties involved.

Solution elements were also created, e.g. an operating system kernel and a communication protocol software package. They were more generic than the design elements to which they could be integrated. Their role can be seen as either replacements of similar solutions offered independently from the design elements, or as filling the technological gaps in the design method in order to make it more self-sufficient. The Osdyn project of the Reagenix period is an example of the former and the development of the toy elevator as the Sokrates demonstration system of the latter. Product-related competence of colleagues was isolated from the code generation competence. Also the competence of most members of the steering group was rather isolated, such as code generation competencies of the firms I and N. Since Sokrates had all the necessary project resources, there was no urgent need for associating the competence of the VTT managers, colleagues, Tekes and the focal researchers with each other either.

Figure 21. The toy elevator built in the Sokrates project.

Table 24. Process-related code generation competence during 1988–1991.

Competence elements/ Owners	**Planned/realised integration of elements**	**Supplier valuation**	**Purchaser valuation**
		(E)xpected, (P)erceived, (H)istorical	
Skills of planning, carrying out and managing contractual code generation projects: focal researchers, (TKO)	Strengthening of internal activity structures/ok Strengthening of actor bonds with Tekes and industry/mostly neutral	Researchers: E+, P +, H +/– Managers: E ++, P+/–, H +/– Colleagues: H –	Steering group: P +/–, H +/– Tekes: H + Evaluators: P +/– Industrial and research customers: P +, H +
Skills of popularising, documenting and publishing R&D results: focal researchers	Strengthening of actor bonds (and activity links) with Tekes, industry and the R&D community /in part competing	Researchers: P +, H +/– Managers: P +/–, H +/– Colleagues: H –	Steering group: P +, H + Tekes: H + Evaluators: P + Customers: H –
Skills of training practitioners and educating students: focal researchers	Strengthening of actor bonds (and activity links) with educational institutes and industry /ok	Researchers: P ++, H ++ Managers: P +, H +/– Colleagues: H +	Industrial and research customers: P + Educational institutes: P ++
Skills of conducting (international) scientific research: focal researchers, TKO	Strengthening of actor bonds (and activity links) with the scientific research community/neutral	Researchers: P +, H +/– Managers: P –, H +/– Colleagues: H – –	Evaluators: P + Scientific research community: P +/–
Parties isolated from the competence elements: Evaluators (Finsoft, Finnish Academy), because of the post-mortem analysis Some members of the steering group Colleagues (except members of the research council)			

The main *process-related competence* element during the Sokrates period involved conduction of the actual research process. It strengthened both the external actor bonds of TKO and the internal activity structures among the Sokrates project group. However, it did not offer notable strengthening for internal resource collections and activity structures or external resource ties and activity links, since the project was quite self-sufficed.

In comparison, the Speco project can be characterised as a close relationship of the focal researchers and the firm K for the purpose of creating and making use of code generation resources. The relationship was co-ordinated and controlled by the VTT managers, interested in the strategic alliance with the customer company. Although the Sokrates project involved a number of actor bonds, it only showed rather few activity links and resource ties. The training on the Sokrates-SA method given by the researchers to industrial professionals and students resulted in bonds with a large number of actors, but not in any considerable activity links or resource ties.

The focal researchers thus became quite distant from external parties with regard to the exploitation of code generation resources, with the exception of the firm Nm. In particular, no activity links were established with the firm N, the code generator expert of which joined TKO, but not the Sokrates project. The Sokrates project team did not totally succeed in making use of technical and managerial documentation to link its activities and their results with the activities and resources of the steering group. The group complained regularly on delayed or missing pieces of information. There was an even more obvious lack of activity links and actor bonds concerning scientific and professional publications and interaction with recognised members of the software engineering research community. Many other Tekes-funded projects had such bonds and some also rather extensive activity links with foreign research institutions. However, the lack was not considered to be too problematic by the interviewed VTT managers.

The evaluation of the process-related competence elements shows some interesting differences. The managers and even the researchers themselves considered the historical value of the actual research results as somewhat mixed. The managers started to suspect the usability of the results even before the end of the project. The researchers explained in the interview that the Sokrates code generator from April 1991 was technically relatively immature and could not yet be used in any serious software development work. This was pointed out by one of the interviewed colleagues as a possible reason for the modest industrial interest in using the results.

However, both Tekes and the early customers were quite satisfied with the results and the way the results had been developed and exploited. Some members of the Sokrates steering group viewed the focal researchers as too self-satisfied, but rather many of them turned out to be just passive bystanders not associating with the researchers by any other way than by attending steering group meetings. The training given by the researchers helped to disseminate information on the results, which was, after all, what most of the steering group members were obviously expecting. Although the lack of interaction with recognised international software engineering researchers was criticised internally at TKO, the steering group and customers did not consider it any remarkable problem either. The foreign Finsoft evaluators indicated that the results were a match for the best international research. The evaluators performed, however, only a post-mortem analysis based on technical documents and a short interview, which can easily lead to misleading remarks.

Many of the Sokrates steering group members were actually also isolated, because they did not master the technology and their needs were not directly addressed in the project. Many of them did not mention code generation as their primary interest. Yet another notable party isolated from the emerging code generation competence was the TKO researchers, who were not involved in other Finsoft projects and therefore did not participate in the research council, which held regular meetings to exchange information on the projects. These colleagues included researchers who had been involved in the initial planning of the Sokrates project, as well as others who used to be members of the R Group, established as an internal interest group.

Table 25. Organisational code generation competence during 1988–1991.

Competence elements/ Owners	Planned/realised integration of elements	Supplier valuation	Purchaser valuation
		(E)xpected, (P)erceived, (H)istorical	
Skills of making use of contacts with Tekes, other VTT laboratories and industry: TKO	Strengthening of actor bonds and webs/ok	Researchers: P +, H + Managers: P +, H +/–	Tekes: H ++ Research customers: H + Industrial customers: H +
Skills of marketing R&D results and making use of the good reputation of the institute: TKO	Strengthening of internal activity structures/neutral Strengthening of activity links with industry/partially ok, Including the VTT Electronics laboratory	Researchers: P ++, H +/– Managers: P +, H –	Steering group: P +, H +/– Tekes: H + Evaluators: P + Industrial and research customers: P +, H +
Skills of organisational planning and project quality management: TKO	Strengthening of internal activity structures/competing Strengthening of activity links with industry/competing	Researchers: P –, H – – Managers: P +/–, H –	Steering group: P +/–, H + Tekes: H + Industrial and research customers: P +, H +
Parties isolated from the competence elements: Management group of the Finsoft program (except J. Karjalainen, H. Hakalahti)			

One of the most important *organisational competence* elements during the Sokrates period was the skills of TKO in maintaining and making use of its contacts with Tekes and industry. This resulted in quite strong external actor bond and webs, although activity links and resource ties were much less common, and internally the Sokrates project team was rather isolated from the other researchers. Moreover, since Jukka Karjalainen and Hannu Hakalahti were almost the only contacts of Sokrates with the management group of the Finsoft program, the program level remained isolated from the project and its steering group.

Some mixed opinions were given in interviews regarding the capability of TKO to market the Sokrates results. The managers started to suspect the marketing plans of the researchers even before the end of the project, regarding them as wishful thinking. The two parties disagreed also concerning the appropriate management of projects. Although the steering group seemed to take the side of the managers in this matter, it did not give remarkably negative feedback in this regard, after all. The group was invited to launch Tekes-supported spin-off projects early, but only the firm Nm used the opportunity. Historically, the managers themselves missed the opportunity of turning the firm N to a key exploiter of the results. They helped the researchers in trying to establish an activity link with the firm Is, but it did not succeed. On the other hand, VTT Electronics laboratory was an unexpected exploiter of the results, which seemed to provide a good case for internal marketing of the results among VTT.

7.5.3 Competence during the Reagenix period

The *product-related competence* elements became quite versatile during the Reagenix period, Table 26. Although the Sokrates generator was not taken in industrial use, the Sokrates-SA method and the system solutions were utilised by industrial and research customers. The value of the former was perceived as high by all parties, but the historical value was seen as remarkable only by the researchers and the firm N which were making successful use of the operating system principle developed in Sokrates.

Although the applications in which the Reagenix code generator was utilised were more diversified than earlier, when the focus had been on machine automation, it was mainly the managers only that suspected the researchers' skills in this regard. Some of the representatives of VTT Electronics laboratory were also critical, while, on the other hand, claiming that the TKO and ELE managers were responsible for the lack of adequate support to the use of Reagenix. However, the focal researchers provided a great deal of help especially for research customers, considering that they were usually not paid for it.

Such broad co-operators as the firm Nm sought for comprehensive embedded software development approaches. This aim did not create particularly strong resource ties with the customers – the projects listed in Table 16 were rather small and did not result in continuing projects. They were still perceived as good, except maybe the Raski project carried out for the firm R, which was disappointed with the results. The VTT managers were not fully satisfied with the way some projects were implemented. The managers were, in their opinion, not extensively enough involved in the co-ordination and control of the projects. The focal researchers thought that the managers did not understand how the code generation technology should be transferred in industrial use in such an efficient manner that would benefit the customers. The firm Is viewed the selling of Reagenix licenses as an attack against its own generator, but did not raise any public dispute over it.

Also internal competition emerged at ELE, concerning Reagenix as "the best approach to embedded systems software development". One of the focal researchers pointed out that it was the "American style marketing that irritated everyone". Although the historical evaluation of MCS-REA results by the firm Nm was devastating, the representatives of the company blamed in part themselves as incompetent buyers, who were trying to make too ambitious a leap. The perceived value of the results had been very high. Focused buyers, be they research or industrial customers, were more satisfied also in a historical sense. The VTT managers did not regard the exploitation of individual tools and system solutions as any great success, though. According to the focal researchers, this kind of viewpoint was totally wrong, the purpose of VTT should have been to help customers as effectively and efficiently as possible, not to make big profits at their cost. Although the individual tools and system solutions were utilised also internally in several research projects, they did not become integrated into any considerable larger resource collections, but remained rather isolated from the other resources of ELE.

Table 26. Product-related code generation competence during 1992–1998.

Competence elements/ Owners	**Planned/realised integration of elements**	**Supplier valuation**	**Purchaser valuation**
		(E)xpected, (P)erceived, (H)istorical	
Skills of applying embedded computer system technologies (and system solutions): focal researchers	Strengthening of resource ties with the industrial and research customers/ In part ok, mostly neutral	Researchers: H ++ Managers: P +/–, H –	Research customers: P +, H +/– Industrial customers: P +, H +/–
Knowledge of embedded systems applications: focal researchers (in part absorbed from customers)	Complementing of resource ties with the industrial (and research) customers/ok	Researchers: H ++ Managers: P +/–, H +/–	Industrial customers: P +, H + Research customers: P +, H +/–
Skills of developing and transferring in use code generation driven embedded systems design approaches: focal researchers	Strengthening of internal resource collections/ Competing Strengthening of resource ties with broad co-operators/ in part ok	Researchers: H ++ Managers: E +, P +/–, H –	Industrial and research customers (broad co-operators): P +/–, H +/– – Tool vendors: P +/– –
Skills of developing and transferring in use code generation based design and testing tools: focal researchers	Strengthening of internal resource collections/neutral Strengthening of resource ties with focused buyers/ok	Researchers: H ++ Managers: E +, P +/–, H +/–	Industrial and research customers (focused buyers): P +, H + Tool vendors: P +/–
Skills of extending code generation functions further: focal researchers	Strengthening of resource ties with customers/mostly neutral, in part competing	Researchers: H ++ Managers: E +, P +/–, H +/–	Industrial and research customers: P +, H +/– – Tool vendors: P +/– –
Parties isolated from the competence elements: VTT managers: did not control the development and application of Reagenix Research customers: only used Reagenix, did not control its development			

One of the main reasons for this might have been that the other researchers were users of the developed solutions, not their active developers. The code generation functions that were made explicit as transformation rules during Sokrates became incorporated inside the code generator tool. There were plans to design code generation functions for several programming languages, but they did not materialise until in the Nosto project carried out for the firm Kc. Therefore, the functional dimension of the competence became less distinct than, for example, in the Kaapeli project where it was explicitly used for the needs of the firm Nm.

Table 27. Process-related code generation competence during 1992–1998.

Competence elements/ Owners	**Planned/realised integration of elements**	**Supplier valuation**	**Purchaser valuation**
		(E)xpected, (P)erceived, (H)istorical	
Skills of planning, carrying out and managing self-funded code generation research projects: TKO	Strengthening of internal activity structures and resource collections/competing	Researchers: P +/–, H +/– Managers: P –, H –	–
Skills of planning, carrying out and managing contractual code generation R&D projects: TKO	Strengthening of internal activity structures/in part ok Strengthening of activity links and patterns with Tekes, industry/in part ok	Researchers: P +/–, H +/– Managers: E +, P –, H –	Tekes: P – Industrial customers: P +, H –
Skills of packaging R&D results into commercial products and IPRs/ focal researchers	Complementing (and strengthening) of resource ties (and activity links) with industry/competing	Researchers: P +, H + Managers: P +/–, H –	Industrial customers: P +, H +/– Research customers: P +, H +/– Tool vendors: P +/–
Skills of training practitioners and educating students: focal researchers	Strengthening of actor bonds (and activity links) with educational institutes and industry/ ok	Researchers: P ++, H ++ Managers: P +, H +/–	Industrial and research customers: P + Educational institutes: P ++
Parties isolated from the competence elements: Research customers: only used Reagenix, did not carry out code generation R&D			

Evolution of the *process-related competence* elements became stalled after Sokrates (Table 27), because of the rapid collapse of the project portfolio. The autonomous position of the researchers in terms of process-related and organisational resources disappeared. The VTT managers and Tekes expected the exploitation of the results to materialise as red projects – far too soon, as was claimed by the focal researchers. The managers were disappointed with the actually realised volume of red projects. They did not prevent the researchers from following their own logic of action, but did not consider the license selling any great success.

The use of the Reagenix code generator in blue projects strengthened internal resource collections, but not activity structures, because the focal researchers were seldom involved in the projects as active participants. The self-funded green research related to code generation partially resulted in competition, not only because the firm Is was developing its own tool environment simultaneously, but also since ELE researchers were investigating alternative approaches in other blue projects. One of the best process-related competence elements was, again, the researchers' skills of training students and customers. However, many of the research customers learned the use of the generator much by themselves.

Table 28. Organisational code generation competence during 1992–1998.

Competence elements/ Owners	**Planned/realised integration of elements**	**Supplier valuation**	**Purchaser valuation**
		(E)xpected, (P)erceived, (H)istorical	
Use of contacts with Tekes, other VTT institutes and industry: TKO/ELE	Strengthening of actor bonds and activity links/in part ok	Researchers: P +, H + Managers: P +, H +/–	Tekes: H ++ Research customers: H + Industrial customers: H +
R&D marketing skills and making use of the reputation: TKO/ELE	Strengthening of internal activity structures/competing Strengthening of activity links with industry/partially ok	Researchers: P ++, H + Managers: P +/–, H –	Research customers: H + Industrial customers: H +
Organisational planning skills and project quality management: TKO/ELE	Strengthening of internal activity structures/ok Strengthening of activity links with industry/ok	Researchers: P –, H – – Managers: P +, H +	Research customers: H +/– Industrial customers: H ++
Parties isolated from the competence elements: Code generation researchers: were not involved in organisational planning and management regarding the code generation competence			

The main elements of the *organisational competence* (Table 28) during the Reagenix period involved making use of the extensive contacts and good reputation of VTT among potential customers, marketing of research and development skills, and planning for and carrying out R&D activities. From the viewpoint of Tekes and customers, the organisational competence of VTT was high. However, the VTT managers and the focal researchers both perceived and indicated in the interviews that they did not value each other's competence very highly in this regard. The use of the Reagenix code generator in blue projects was perhaps an exception, but otherwise the two groups of actors had completely different viewpoints. This is clearly visible in the two intertwined case stories included in Appendix 2.

As a result, the focal researchers actually became isolated from the organisational management of the code generation competence. As an example, the plans to establish the small company that bought the intellectual property rights of the code generation technology may have in part resulted from inquiries to me and one of the ELE group managers from a third party, regarding the possibility to buy the rights. As another example, the use of the generator in blue research projects was planned and co-ordinated by the research customers themselves, not much by the focal researchers. The co-ordination of this internal exploitation was based on informal personal collaboration, not on any organisational plans.

8 ANALYSIS AND IMPLICATIONS OF THE RESULTS

In the following I will discuss how well the competence evolution framework, as it is summarised in Figure 17 in Chapter 6, manages to explain the development of the code generation competence. I will also compare the code generation case with the fault diagnosis case discussed in Chapter 5, by evaluating the competence and relationships involved in the two cases. I will close the chapter by summarising the main empirical, research and managerial implications of the code generation case study.

8.1 ANALYSIS OF THE COMPETENCE EVOLUTION

The purpose of the competence evolution framework is to help understand the development of competence, i.e. the changes of activities to create and make use of resources, in terms of the development focal nets. The activities pursued by certain actors are a means of implementing their logic of action, when aiming at reaching goals that fulfil specific objectives.

[Klein et al. 1998] is a recent example of the so called competence charts used for illustrating "clusters of R&D skills". Klein, Gee and Jones are proposing skills consisting of technical, processual and human aspects, which correspond to the product, process and organisation-related competence elements of this research. Their "company skill maps" are used for matching process-related competence elements to product-related elements in different organisations. An evaluation of skill levels is also included. Mutually supportive skills can be associated with each other, so as to form "skill cluster diagrams" representing, according to the authors, the core competence of the organisation.

Instead of this kind of an intra-organisational competence chart, Table 29 shows code generation related product, process and organisational competence elements of both VTT and the most influential external groups of actors – as a result of their logic of action that is indicated in the table.

During the Speco period, a vision of code generation was developed and tested. The vision was, however, mostly created by the firm K. VTT implemented it, the customer bought a technical solution of the vision, a prototype of a code generator tool. The generator was implemented by using the programming language Prolog that was familiar to both VTT and the firm K. However, the opinions on technical matters of the customer and the focal researchers started to differ, not to speak of the process of innovative research. The functional skills of the researchers were related to compilation of program languages, to managing concurrence, as they explained in the interview, whereas the customer was familiar with the AI reasoning functions. The kinds of automation systems that the firm K developed were considered key embedded systems applications by both parties.

Table 29. Code generation competence 1985–1998.

Actors/ Logic	**Product-related elements**	**Process-related elements**	**Organisational elements**
Speco (1985–1987) – Testing of an industrial vision: strategic red alliance			
VTT: Project business	<u>Design elements</u> Applications: (automation) Functions: compilation Techniques: programming Technologies: Prolog	Innovative research Project marketing and implementation	Strategic alliance R&D groups Domestic reputation Utilisation of Tekes Technological skills
Firm K: Solution buying	<u>Design elements</u> Applications: automation Functions: AI reasoning Techniques: programming Technologies: Prolog	Development of an innovative integrated embedded software engineering environment	Company vision R&D groups Corporate research Utilisation of Tekes Automation business
Sokrates (1988–1991) – Demonstration of a research vision: joint blue national effort			
VTT: Pioneer-ing research	<u>Design elements</u> Applications: automation Functions: code generation Techniques: Sokrates-SA Technologies: generator <u>Solution elements</u> Technologies: SIC protocol package, ReagOS	Technology development Applied research in blue (red) projects Training of Sokrates-SA Technology popularising	Managerial role in the Finsoft program Utilisation of Tekes
Firm Nm: Solution buying	<u>Design elements</u> Applications: automation	Product development	Machine automation business
VTT/Ele: Faster solutions	<u>Design elements</u> Applications: automation, VLSI design	Applied research and development in blue and red projects	Utilisation of Tekes and industry R&D groups
Reagenix (1992–1998) – Fighting for the vision: missing red R&D business			
VTT: Product business Cheaper solutions	<u>Design elements</u> Techniques: (Sokrates-SA) Technologies: generator, testing tool, animator <u>Solution elements</u> (cf. above) (Specific competence in blue projects)	Applied research and development in blue and red projects Use of blue project resources	Utilisation of Tekes, industry and other VTT institutes R&D groups Utilisation of Tekes and industrial contacts
Firm Nm: Paradigm buying	(Cf. above)	(Cf. above)	(Cf. above)
Firm R: -"-	<u>Design elements</u> Applications: electronics	Product development	Consumer electronics business
Firm S: Solution buying	<u>Design elements</u> Applications: electronics Functions: software design	Product development In-house design of software tools	Consumer electronics business (R&D groups)
Firm N: -"-	<u>Solution elements</u> Technologies: embedded software and hardware	Product development In-house design of software tools	Telecommunication business R&D groups
Firm Kc: -"-	<u>Design elements</u> Applications: automation	Product development In-house tool design	Machine automation business

Regarding the outer context, with increasing industrial interest and active planning of joint national software engineering research, favoured the utilisation of the organisational and project planning skills of TKO for making use of the technical results of Speco and Draco in marketing Sokrates. The role of the firm K cannot be neglected either, as its corporate research unit, for which the Speco project was carried out, was a forerunner in embedded software engineering and had close contacts with Tekes. Although the funding organisation was in a critical position regarding financial resources available to applied engineering research, its role in technology development and management was rather low. This was paving a way for the autocracy of the focal researchers in Sokrates, where they could follow their own logic of action and had all the necessary technical, human and financial resources to make their own vision true.

During the Sokrates period, the core concepts and implementations of code generation design elements were indeed produced, in addition to a few unforeseen solution elements. A remarkable difference with regard to the results of Speco was that these elements were more generic. The Kaapeli and Synchro projects in which the results were applied for the first time involved automation applications, and the Sokrates-SA system design technique, the code generation functions and the implementations of the code generator and solution elements were portfolio-specific. The understanding of the focal researchers on automation applications helped them to co-operate with the firm Nm and VTT Electronics laboratory. VLSI design was new to them as an application, after the Sasic project VTT Electronics laboratory carried code generation further with the University of Oulu, without involvement of the focal researchers. They claimed that VTT advertised the results of this co-operation more than the Sokrates results.

The Sokrates-SA method was the dominant product-related competence element of Sokrates, but the functional code generation skills also appeared as a useful purchase for the firm Nm in the Kaapeli project. The generator itself was still immature. The opportunity window for producing an industrially usable generator had been closed already. The firm N had such a generator, a very simple one. The firm Is had apparently already started developing its own generator, although it did not tell this to any external parties. The focal researchers and VTT managers could not launch any red projects involving generator development, although the possibility of Tekes-supported spin-off projects was actively marketed to the Sokrates steering group. This was a rather fatal failure, since such a work still needed to be done after Sokrates, but in a much more complex and financially limited setting. According to the researchers, some companies had "learnt too much" and did not need VTT any longer.

The process-related competence was considered to be good by the focal researchers during Sokrates, due to their emphasis on innovative, result-driven and pragmatic research. Some other parties were complaining about the lack of managerial competence. However, the competence was good enough for carrying out the Sokrates project, and it was supported both by the key managerial role of TKO in the Finsoft program and the excellent training and technology popularising skills of the focal researchers.

While Sokrates was kind of golden time for developing code generation competence, *the Reagenix period* brought the secure focal net harshly down. The promise of extensive exploitation was, from the viewpoint of the VTT managers, rapidly fading away. The self-controlled huge project resource pool and the central position in the large Finsoft program were no longer available to the researchers. The large Sokrates steering group had dissolved and regular interaction with parties which had been interested in code generation for the past three years, was finished. Most importantly, it was now much more difficult for the VTT managers and the researchers to unite their forces to market the competence than it had been to launch the Sokrates project. The managers pursued the goal of initiating large enough red projects, whereas the researchers started to realise that the code generator needed to be improved, if not developed anew, without too much resources or external support, and facing competition from the firm Is.

It seems that the increasing non-alignment of the logic of action of the researchers and managers had evolved little by little. The managerial problems of the Speco project and the dissolution of the R Group may have left some tension between the parties, but as the managers pointed out in the interviews, they actually believed in Sokrates more than some of the colleagues of the focal researchers did. The idea of starting product business at VTT was tolerated by the managers, who even defended VTT's interests towards the firm Is on a few occasions. However, after the financial outcome of the business turned out to be modest, their interest in the matter collapsed. The focal researchers thought that the managers were utilising customers rather than supporting efficient technology transfer. At the group rehearsal, they were strongly criticising the managers' distrust on the skills of VTT to develop useful software engineering tools for industry. The managers were claiming that the researchers themselves had missed the tool development opportunity during the Sokrates project already.

The diversity and small size of red projects during the Reagenix period is apparent also in Table 29. Some customers were interested only in certain solutions, others in more comprehensive offerings. The firms S and N, which had developed software design tools for internal use, did not, after all, make use of the design elements related to code generation. The role of SA/SD was decreasing, which according to the steering group members affected their interest in using the results. The functional code generation skills remained hidden inside the generator, which made their exploitation difficult without the tool technology. Blue research projects seemed to offer alternative networks for exploitation, and the corresponding experiences from Sokrates were encouraging. However, the projects became a sort of extra front of colourless relationships, where the researchers spent their time without any considerable rewards. Although this kept Reagenix alive, it did not help either in the project sense or product business sense to improve the organisational status and visibility of code generation. The same holds for licenses that were sold or given for free to external parties. After many of the focal researchers had left VTT, the effective informal network that had helped to make use of Reagenix in blue projects weakened. Selling the code generation technology rights to the researchers was obviously an appropriate move, carried out at the last minute.

8.2 CROSS-CASE ANALYSIS

The fault diagnosis case involved VTT, but mostly a different group of actors and another area of competence, which was industrial fault diagnosis systems. The framework that was used to make explicit the evolution of fault diagnosis competence and relationships did not yet include the concepts of logic of action and processes applied in the revised framework. I will therefore discuss them briefly in the context of the fault diagnosis case, as depicted in Table 30 using competence elements, processes, actors, logic of action and project nets. The two cases are then compared from the viewpoint of competence evolution, based on the alignment of the logic of action of different parties and change of the relationships.

Table 30. Evolution of fault diagnosis competence and relationships.

Period	**Competence Elements**	**Dominant processes**	**Actors/Logic of action**	**Project nets**
1986–1988 *Dawn of intelligence*	Product: AI and KE techniques	Green research	VTT/Establish a new research area at TKO	Mostly informal actor bonds
1989–1991 *Demon-stration of knowledge-based systems*	Product: KE techniques, tool technologies Process: system design skills Organisational: knowledge engineers	Marketing of the promises of KE applications Blue and red projects: design of system prototypes for industrial trial	Process and machine automation firms and VTT/Improve the skills of knowledge engineers	Innovative project nets Informal interaction of knowledge engineering experts
1992–1994 *Develop-ment of reliable automated products and processes*	Product: fault diagnosis functions, automation applications, KE techniques Process: system design skills Organisational: R&D projects	Marketing of fault diagnosis solutions Red and blue projects: development of diagnosis functions for automation applications	Machine automation firms/Improve reliability of products and processes VTT/Develop intelligent fault diagnosis systems	Product and process development project nets Informal actor bonds between diagnosis and automation experts
1995–1997 *Extending of the bridgehead*	Product: fault diagnosis Platform, KE techniques Process: fault diagnosis system design Organisational: References, R&D projects	Marketing of a core platform Analysis and capture of new R&D markets Green, blue and red projects: development and use of the core platform	Automation and telecom firms/ Manage complex systems better Funding bodies/ Support R&D of intelligent systems VTT/Open up new application areas	Industrially-driven R&D project nets Informal actor bonds between diagnosis and application experts (and end-users)

Goals and logic of action

The goal of developing code generation as a means of helping professional embedded software engineers emerged during the Speco period. VTT, which had formed a strategic alliance concerning the development of an advanced software engineering environment with the firm K, could use the technical results of the Speco and Draco projects and its organisational skills for creating a pivotal position in Finsoft. Tekes and industry bought the promise of code generation marketed by the VTT managers and researchers. The other parties let the researchers take the lead, because Finsoft addressed generic techniques rather than industrial applications.

The researchers had already formed a core problem solving team and were applying their radically new technical ideas using a well-known domestic CASE tool and the popular SA/SD method. This facilitated the acceptance of the idea, as did also the experiences reported on the Speco (it is possible to develop practical code generation technologies) and Draco (existing solutions are not practical enough or even working in embedded systems applications) projects. The code generator designed in Speco was thrown away to start the development again, this time based on the researchers' own vision and solutions. Even the firm K had trust on the researchers, although it had been complaining about managerial problems in the Speco project.

The starting of fault diagnosis related R&D activities at TKO, characterised as the "Dawn of intelligence" in Table 30, bears some resemblance with the Speco period. In both cases, a new research area was opened by scanning existing approaches and technical solutions. However, the vision of the role of fault diagnosis systems in industry was very weak, if explicit at all, compared with the Speco and Sokrates visions. The vision was gradually created during 1989–1991, together with a few industrial knowledge engineers. A small team of VTT and industrial researchers started focusing on the so-called embedded expert systems at that time, which closely corresponds to the co-operation between the firm K and TKO during Speco.

The industrial people involved in the early phase of fault diagnosis R&D were knowledge engineers, who wished to strengthen their own position and to improve their skills in applying knowledge engineering techniques. This resembles the relationships between the focal researchers and the representatives of the firm K in Speco. The early fault diagnosis R&D results were considered demonstrations, rather than meant for daily industrial use. This was the status of the Speco results, too, and it had apparently also been the aim of the project.

Generic techniques and novel technologies were developed and applied to solving problems both in particular code generation and fault diagnosis applications. This kind of marketing and demonstration of the promises of new technologies for specific customers in the context of some emerging or already established engineering techniques is considered one of the core activities in contractual engineering R&D – much along lines of the logic of action followed by Tekes, as en engineering research funding body.

In both cases, the logic of the two key actor groups of the focal organisation, the managers and the researchers, were well aligned. They were marketing and exploiting the initial customer projects together, so as to pave the way for subsequent competence elaboration and extension. In code generation, one big blue project was established, while in fault diagnosis several smaller red projects emerged.

Change of project nets

During the Sokrates period, the focal researchers were enjoying a great deal of autonomy from the other parties thanks to their technical skills and the huge project resources. They were controlling the extension of the core problem solving team by themselves, hiring mostly newcomers, who from the viewpoint of VTT colleagues started to think and behave "the Sokrates way". This kind of human code generation skills management left a lot in the hands of the researchers. For example, the managers did not use their organisational authority for allocating the former code generation expert of the firm N to the Sokrates team. From the managerial viewpoint, this could have greatly helped to exploit the company as one of the key customers of TKO, who had developed similar solutions. The coming rapid growth of the company's business sector, telecommunications, was also not foreseen. Since the exploitation of the results at the firm Nm and VTT Electronics laboratory had started, industrially useful results seemed to realise anyhow.

Compared with the code generation case, in fault diagnosis the technology research and demonstration phase was much shorter and more fragmented. One of the critical differences can be found in the fact that fault diagnosis, as opposed to code generation, soon drew away from the generic AI techniques and special tool technologies upon which it had originally focused. By addressing fault diagnosis as a system design problem rather than an AI application, the VTT researchers could make use of a broader set of embedded systems design techniques and tools, which made the basic principles of fault diagnosis understandable to customers.

This was necessary, as fault diagnosis functions were incorporated into industrial systems and processes designed and maintained by other people than knowledge engineers. The portfolio of automation applications helped the researchers to increase their application skills and understand the technical experts on the customer side. This focus was kept for a rather long time, by directing project marketing activities to the automation sector and using past projects as references. Despite, or perhaps due to, the recession, automation companies were investing in reliable products and processes, which created room for fault diagnosis systems. Although the former relationships of VTT with industrial knowledge engineers were broken up, new relationships with automation engineers and end users were created. Both the VTT managers and knowledge engineering researchers wished to establish red fault diagnosis projects, while the reliability requirements of the users of automated machines and processes were increasing the pressure on product manufacturers. The vision of knowledge-based fault diagnosis and its implementation could thus be carried further without any considerable disagreements within VTT and in its customer relationships.

Evolution of competence

One company disappointed with the results of a red fault diagnosis project for which it had contracted from VTT, was an unusual exception. In other projects, the reuse of fault diagnosis functions, the use of known techniques for modelling automation applications along with a variety of implementation technologies of the customers helped VTT to create sustainable portfolio-specific competence and allowed it to remain quite flexible at the same time. This resulted in the evolution of a core platform, around the functional product-related competence dimension. It opened up both opportunities for capturing new markets and applying new generic techniques. A problem solving process also emerged, which tied product-related competence elements together.

Sokrates also resulted in portfolio-specific design competence. The Sokrates-SA method was well received by industrial professionals, and the generic system solutions provided with unexpectedly good spin-offs. Although the system solutions and the Sokrates-SA method could be exploited very much as such during Sokrates, they provided only little and sporadic income for the focal organisation. The generator that had been developed was, again, thrown away, and a new one was produced in a few months. However, the use of the Reagenix account that was thought to give the researchers financial freedom to carry out their business logic based on the new generator product, did not succeed as planned. The relationship with the firm Is broke up, and no alternative tool vendor partners could be found. The managers lost their interest in ensuring financial resources for further applications and development of Reagenix, while they were extensively involved in expanding the markets of fault diagnosis systems. The core fault diagnosis platform made it possible to direct competence marketing to the telecommunication sector, as soon as its enormous increase became apparent. No internal research customers were sought for.

A national research program of Tekes was utilised to help to create the first telecommunication fault diagnosis project, but so that Tekes funded a few customer companies which contracted a red project from VTT. This helped Tekes to indicate the usefulness of its funding with regard to transferring new technologies to industry. The fault diagnosis skills were solely owned by VTT, but the companies took care of project management. Problems similar to Sokrates arise, the case application was complex and difficult to model, the first results remained a demonstration system and difficulties were encountered with some techniques chosen to be used. However, the first large red project was followed by two new projects. One of them was carried out for one of the initial customers, the other for a new customer. An extension of this bridgehead was supported by a rather large green project funded by VTT. In 1998, a marketing manoeuvre was performed among instrument manufacturers, one of them getting involved in a green project, and the first red fault diagnosis project for this customer was initiated.

The management of competence evolution through reuse and renewal of the core platform, associated with the creation of customer relationships in new application areas thus seemed to work, if the application knowledge could be modelled and no major mistakes were made in developing or taking in use new generic engineering techniques and implementation technologies.

There were some product-based business opportunities involved also in the fault diagnosis case. For example, a distinct tool was developed for explaining faults detected by diagnosis systems. These opportunities have not yet been used, and some of them seem to have been lost due to their use in red projects. At least one customer has sold the rights of a distinct fault diagnosis competence element to a third party without any consultation with VTT. Although this is completely correct in legal terms, it indicates that VTT has not been able to manage all the aspects involved in its competence. If competence elements are packaged well enough, the purchasers may re-sell them to other parties. This is possible, when a customer becomes the owner of project results based on reuse of explicitly codified competence elements. From the R&D supplier's viewpoint this involves the problem of "satisficing" codification and protection of its core competence.

Effects of the outer context

Interestingly enough, as mentioned in passing above, the developments in the outer context during the period 1990–1995 apparently affected the two cases in opposite ways. The recession that considerably decreased the role of the automation sector as a potential application field for embedded software engineering techniques, seems to have boosted the need for developing more reliable products. This helped VTT to create portfolio-specific fault diagnosis competence with automation companies. The buyers and end users of expensive automated machines and processes wished to make use of their investments efficiently, and the hard times forced machine manufacturers to listen carefully to their customers' needs for reliability.

Since many manufacturers did not have the capability of building advanced computerised fault diagnosis systems themselves, they were willing to use external experts. VTT researchers had developed an approach to fault diagnosis, in which engineering models familiar to manufacturers could be used. Intelligent techniques were applied in developing effective fault diagnosis functions and any technologies favoured by the customers. Relationships could be established where the interests and competence of different parties matched. One customer was clearly dissatisfied with the competence: "The results were used in a continuing project, where the approach was used to collect and analyse data acquired during six months from nine different … [products]. I am more satisfied with these results than the first project. Otherwise the results could hardly be utilised."

If the code generation competence had been matched with the developments of the outer context, it would have been possible to make explicit the functional competence elements and to associate these with the kinds of new system design techniques that were emerging especially in the telecommunication sector.

Towards a code generation core platform?

Code generation research started from a rather technological origin, which is understandable when considering the background of the key researchers. Most of the early embedded systems applications involved "low-level" device control software, but in the nineties the focus was directed to "higher-level" applications and software architectures providing some added value to the basic device control. System design techniques and tools should therefore help to develop and reuse software architectures, not only the kind of low-level device control software embedded, for example, in the toy elevator built in Sokrates.

Based on the experiences gained from the fault diagnosis case, Figure 22 illustrates how code generation competence might have been packaged into a core platform. Since electronic products and systems have in the nineties evolved towards technical, functional and application platforms built much one on top of another, the functional code generation competence could have been associated with different design techniques and implementation technologies used at different platform levels. Even the researchers concluded in the group rehearsal that software architecture concepts created during Sokrates were not utilised well enough.

As an example, the device control platform of a digital telecommunication product involves digital signal processing software, the functional platform communication protocol software, and the application platform user interface software. Focusing on a software platform architecture might have led to the creation of relationships with different customers or different experts of the same customer would have been required, similarly to the development of fault diagnosis systems at ELE.

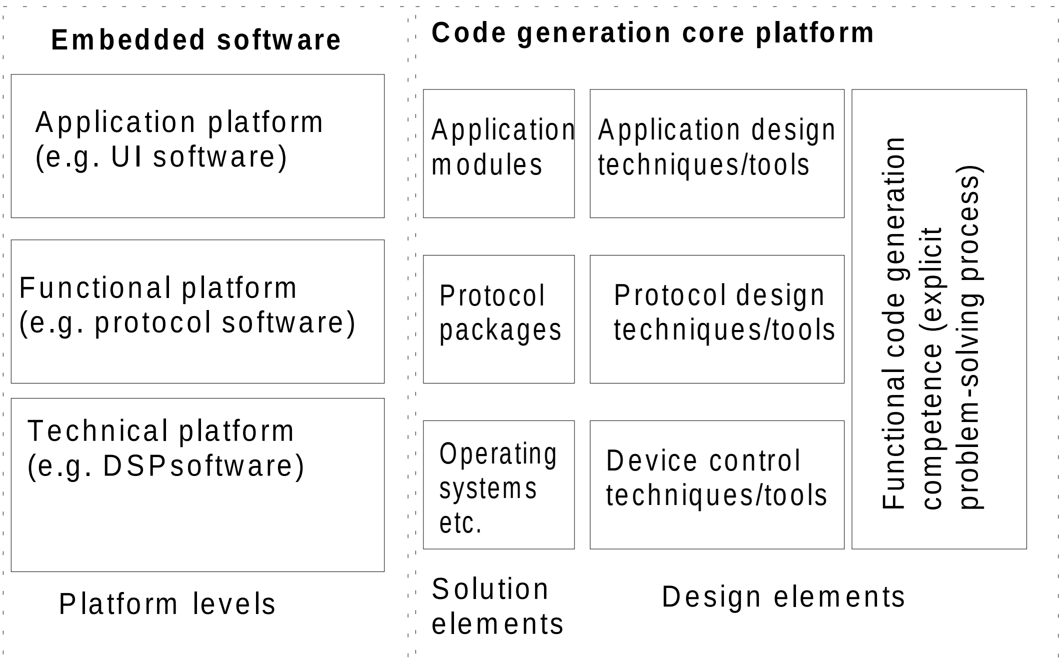

Figure 22. Embedded software platforms vs. code generation platform.

8.3 LESSONS LEARNED

The research-related, theoretical and managerial findings of my study are discussed in the following, focusing on lessons learned for subsequent investigations of competence in the context of industrial networks.

8.3.1 Insights for research

I found the role of secondary data quite crucial in the code generation study. Although secondary data may be overwhelming in longitudinal studies, digging deeply into it seems to pay off. In this research, the data played a central role in making explicit both the project nets and processes involved in the code generation case. I used the competence evolution framework to structure the data into meaningful chunks during successive periods of time between 1985 and 1998. However, it was rather difficult to prevent the secondary data from becoming too fragmented, because of all the details included in the documents from which the data was acquired. For this reason, the data was organised around the competence marketing and purchasing, R&D, competence exploitation and mutual co-ordination processes. This helped much to follow the evolution of competence.

During the research, it became obvious than an important piece of secondary data was missing – information related to the outer context of the case phenomenon, not only from such a limited perspective as the documents describing the Finsoft research program that I had included in the data sources, but from the viewpoint of environmental macro forces affecting industries and markets as a whole. I decided to acquire the required data on the outer context, which was actually used by the managers for the purpose of positioning VTT in the changing environment.

This proved a rather good approach, because I managed to identify certain pieces of industry-level information that had remained unnoticed and, therefore, could not be considered in guiding the activities of the focal organisation. One of the best examples was the strong bias shown by TKO towards automation applications in embedded software engineering at the turn of the nineties, when the existing R&D potential of the sector was quite modest and rapidly decreasing, and the telecommunication sector was about to start its immense growth.

The secondary data does not include too much information on why certain actors took part in certain nets and were involved in certain activities. Although some of the key expectations and perceived values of the activities and their results were summarised in managerial documents, no data on the longitudinal or historical effects of the code generation competence was available. The evaluation of resources used in or produced by R&D activities is a topic constantly under debate, while almost invariably only the situational values are addressed. Various kinds of frameworks and criteria have been proposed for measuring the R&D process and its outcome, for example [Cooper and Kleinschmidt 1996, Roberts 1995, Schumann et al. 1995, and Tipping et al. 1995].

Schumann and others have specified a set of “total quality measurements” for R&D, Tipping and others a comprehensive “technology value pyramid”. [Werner and Shouder 1997] provide an exhaustive survey of the R&D measurement literature from 1956 to 1995. To compare, Holmlund [1997] proposes technical, social and economic dimensions for relationship quality. The content of the technical dimension of relationship quality proposed by Holmlund includes generic process types and characteristics, as well as technical outcome. The social dimension of quality includes the individual level and the company level, the economic dimension the costs and benefits of the relationhip. No distinction is made between the expected, perceived and historical value of relationships. The secondary code generation case data included information on the expected and perceived values of the competence, the historical value was captured from the primary data. The primary and secondary data included a number of interesting and even dramatic differences in this regard.

Parts of the primary data were compiled into intertwined views of the two main groups of actors of VTT. They provided both a summary of the case abstract enough for readers not familiar with the subject matter and an exciting history of the developments affecting the participants, which had never been put together earlier. Compared with focused interviews with the individuals involved in specific activities, such as customers of certain projects, the summaries open up a more comprehensive perspective to the case as a whole. The conflicting viewpoints of the managers and focal researchers became very obvious. Several parts of the original summary written by me were pointed out as "wrong", misleading or incomplete.

The help of information technology, especially electronic mail, was surprisingly beneficial in the process of gathering primary data. The interviewees were very open and willing to clarify their opinions. Some of them had been waiting for this opportunity in vain, which shows how little attention is paid to the past even in organisations aiming at transferring in use knowledge providing long-term value for their customers.

I targeted the bulk of interviews on purpose at persons who had carried out other than only managerial tasks. Although this focus certainly depends in part on the technical nature of the case, I wish to emphasise that business is not only what managers vaguely remember to have happened, but what their subordinates have personally done and experienced. This focus was important also because of the fact that very few informants used the opportunity of commenting or correcting the data that was acquired and documented by me as part of this research.

As opposed to using general managerial data, it is much easier to triangulate detailed data given by people who have carried out certain activities themselves. The size of the interview data of the code generation researchers was about ten times larger than the data provided by the two focal VTT managers. Moreover, the data that for example individual Sokrates steering group members provided was typically less than one page.

8.3.2 Theoretical findings

Analysis of the code generation case addresses the development of competence in one focal organisation as a result of its external relationships (and, to a lesser degree, due to its internal activities). Therefore, I wanted to use the existing research results of industrial networks as a means of explanation rather than as the subject of the research. Accordingly, I focused on testing and applying the relationship-based competence evolution framework proposed in [Seppänen et al. 1998a]. However, the framework needed to be extended by the concept logic of action, which was associated with the network concepts of actors, resources and activities.

Although investigations of industrial networks have utilised research methods familiar from social science and organisational studies, such as unstructured personal interviews, they usually focus on organisations rather than on individuals as research subjects. I chose the logic of action as a means of making a difference between groups of actors when associating them with relationships. It appeared to be a useful way to describe and explain the activities that were pursued by certain groups of actors within the context of certain focal nets. For example, the basic dilemma between product-based and project-based exploitation of the code generation competence could be made explicit and analysed.

However, further research is needed to gain a better understanding of the behaviour of organisational groups of actors in complex network settings. The interaction of "sustaining" and "radical" logic of action would make a particularly interesting research topic. As an example, in the group rehearsal one of the focal researchers compared the aim of using code generators in software development to the revolutionary idea of taking photographs instead of painting portraits at the end of the last century.

One of the great challenges in behavioural studies of network actors is that the focus may span from individual influential persons, such as the manager of a specific project, to whole organisations, industries and even nations. For example, the "logic of Finland" caused the recession, which had a strong influence on the developments visible in the code generation case. Tiittula [1994] uses the concept of "managerial logic" when analysing organisational changes at VTT as a whole. Consequently, this kind of extensive intertwining of inner and outer contexts can be regarded as an essential issue in real-life industrial networks.

Perhaps a more demanding task than the use of the logic of action was, after all, the use of the competence evolution framework as a whole. Although the network-based approach to industrial marketing is deeply rooted in the resource-based perspective to companies, there are only a few in-depth studies on the role of resources in networks. As an example, the need to define the change of the value of a resource over time became obvious during this research, while there was little guidance available on how to treat this issue. Although I did not concentrate as much on the change of the value of resources than on the processes that caused such changes, there is an obvious need to address this problem in further studies.

I used [Tikkanen 1997] as a starting point in the process perspective on networks, while still had to elaborate his approach. In my case the well-archived secondary data, fortunately, made it rather easy to follow the processes during the past years. As indicated by Tikkanen, this may be much more difficult in cases in which such data does not exist, e.g. due to the fact that the relationships are more informal and sporadic in their nature, in contrast to projects lasting for months or years. Despite possible difficulties, I would like to urge network researchers to keep a closer look on processes, because otherwise long-term phenomena may become trivially classified into activities, without the longitudinal "process flow" becoming explicit. Conceptualising and studying this flow is essential when analysing, for example, the true evolution of relationships against marketing plans and strategies. I defined a very simple process flow, considering structures of a few key activities as processes. Dubois [1994], for example, could be used as a starting point to define much more elaborated process frameworks.

While strategic marketing used to be a popular school of thought in the seventies, network researchers have been focusing more on individual companies than on markets as macroeconomic entities. It is therefore noteworthy that the view on product-related competence has been derived from the strategic marketing concept of Abell [1980]. The "techniques" dimension that I have added to Abell's market model is justified by the engineering and scientific foundation of the services that research organisations are offering to the market. I have also used another key concept of strategic marketing, the life cycle, and associated it with competence elements. The usefulness of the life cycle concept is illustrated, for example, by the need of making explicit the simultaneous rise of telecommunication applications and the fall of structured system design techniques in the nineties, which greatly affected the code generation case.

8.3.3 Managerial implications

The use of the market as a mirror of the content of product-related competence leads to one of the main managerial implications of the code generation case: strategic management of competence must be carried out *within* the context of external relationships, not only *for* the relationships. Therefore, the strategic management of the relationship portfolio of a company is actually one of its main competence management tools.

Although self-funded green projects are an important asset for VTT, competence evolves or withers much depending on the organisation’s skills to market and co-ordinate red projects. Differences between the fault diagnosis and code generation cases discussed above indicate this clearly. To put it simply, the code generation researchers considered the market consisting not of red projects, but of domestic CASE tool users, but failed to carry out strategic competence management at VTT after the Sokrates project had been finished. The VTT managers who considered the market as consisting of red projects, basically gave up the competence after the promise of such projects was not fulfilled. The competence survived, fortunately, but withered seriously at VTT.

However, both the managers and researchers seemed to have misinterpreted or neglected the coming industrial changes. Automation and electronic instruments were rapidly losing their customer potential, whereas telecommunication started to emerge as a killer application. Regarding this strategic market change, the decision to allocate the code generation expert of the firm N who joined VTT to a completely different network of relationships in the middle of the opportunity window was a failure. In more general terms, the versatility of code generation competence may have resulted in losing sight of functional skills, which was identified as a fundamental competence dimension in the fault diagnosis case. In the code generation case, such skills involved knowledge of how to systematically produce computer programs from higher-level design models. These skills were hidden within the code generator technology, which was constantly changing and difficult to understand by the likely customers. Although this kind of mechanising of skills was one of the key ideas of the focal researchers, the customers were apparently not ready for it.

The focal researchers claimed that the initial case summary written by me was missing the point of functional skills of managing concurrence in embedded systems, describing the work "just as the development of a yet another CASE tool". This was, however, the opinion of at least one of the customers, whom I interviewed. He regarded the focal researchers' claim of focusing on concurrence management as harshly as "bullshit".

The Kaapeli project, in which the functional skills were made explicit through manual coding rules without any tool technology, was quite successful also in historical evaluation. The MCS-REA project, where the skills were implemented in connection with the Reagenix code generator, was afterwards considered a business catastrophe by the customer, despite that the perceived value of the technology was seen as excellent. Although the SA/SD skills helped to create and to extend the code generation relationships during the Sokrates period, they did not offer any foundation for strategic long-term management of the code generation competence. Instead, the skills were lent core rigidity [Leonard-Barton 1992] that prevented a timely response to the change in design techniques.

Many people were trying to point out this to the focal researchers, but their first love was too strong to be replaced until the late nineties. In the group rehearsal, most of the researchers were still of the opinion that SA/SD was the right choice, and the same was indicated by various other interviewed parties. However, interestingly enough, one of the focal researchers told, as his present opinion, that SA/SD was *not* the right choice.

The relationships between a research institute developing engineering methods and tools and a commercial tool vendor appeared rather problematic in this research. The whole idea of the Finsoft research program was to study and transfer into industrial use new software development methods and tools. However, the co-operation between researchers and industry wishing also make business out of the results was obviously not addressed well enough either in the research program or by VTT.

TKO and the firm Is did not succeed in clearing up the situation. The matter was still sensitive when this research was carried out, as the representatives of the firm Is told me that they did not want to think back to the past events related to the code generation case. For this reason, they turned down my inquiry to interview them. All the other parties were willing to participate and were quite open in the interviews, even the VTT managers and focal researchers who had still clearly conflicting viewpoints regarding each others' goals, activities and work-products.

The focal researchers were quite united in their view that the VTT managers were preventing them from following their logic of action and making business by selling tool licenses. This was obviously true in the sense that the whole portfolio of R&D activities at TKO and ELE was based on the logic of creating red projects from green research via blue projects, instead of focusing on collecting IPR fees. The researchers were not prevented from following their logic, but not much supported either. Since all the organisational and process-related resources were controlled by the managers, the researchers were doomed to fail in their attempt.

The researchers somehow did not see the project portfolio as the basic organisational means of developing the competence. One of the best examples of this is that I, as a lower-level manager who was closely involved in the code generation activities, thought honestly in my original case summary that Reagenix was redesigned on the basis of the Sokrates code generator, following the project portfolio principle. The focal researchers reacted strongly against this view: "What was worst in this story was the indication that the Reagenix code generator was the same as Sokrates. My opinion is that Reagenix has been developing slowly here and there, and there is no clear link between Sokrates and Reagenix". In the fault diagnosis case no developments took place "slowly here and there", but in specific projects conducted jointly by the managers and researchers.

It is interesting that the whole idea of project marketing seemed to be rather distant, if not terrifying, to the code generation researchers: "I did not have any appropriate contacts and I had not received any positive feedback on my work [on Reagenix], so I was not interested in marketing". Many of their colleagues were continuously co-operating with the managers to create contacts, market projects and to share successes and failures in managing evolving competence. The opinion of the focal researchers that the VTT managers only wanted to milk customers – by selling work as projects with no unified frame, some projects addressing "parts of Lada" and others "parts of Porsche" – is radically different from this kind of collaboration.

Despite the frustration and disappointments involved in the code generation case, as in many real-life phenomena, it is relieving to see that the competence survived the difficulties. This must be credited to the true inventors and owners of the competence, the focal researchers. Therefore, one of the managerial lessons that can be learnt from the case is that the commitment of people should be taken better care of for the benefit of competence, whether it is managed according to any logic of action.

An appropriate closing to this chapter will be provided by a small software extract, a piece of code generated with Reagenix shown in Figure 23. The program has been generated for a small demonstration application – this time it is a simulation of a traffic light control system. After all, the light is now green for the generator, which means "go" for profitable business.

```
/* xtlight1.c – control_traffic_lights () – 1997-06-17 13:07:10 */
/*-----------------------------------------------
  Diagram Information
  Title : control_traffic_lights
-------------------------------------------------
 ReaGeniX Programmer Version Version 2.Beta-01
 Licensed to: …
 ReaGeniX is a trademark of VTT Electronics, Finland
-----------------------------------------------*/

/* reactime compatibility check */
#define reactime_style 1
#include "reactime.h"
#if reactime_level < 1
 #error Old reactime version included
#endif

/* subprocess compatibility check */
#ifndef no_flowcheck
 #include "xtlight1.v"
#endif

/* data definitions */
#include "xtlight1.h"

process_body(control_traffic_lights)
 #ifdef flag_init
 initial_value_reservation(initial_value(flag),flag) =
  {flag_init};
 #else
 #ifndef flag_uninitialized
  #error Uninitialized state variable of type flag
 #endif
 #endif

 initialization
 init_process(sequence_lights,C2);
 init_process(register_pedestrian_request,C1);

 init_phase_2
 link_own_flow_from(C2.ped_grant, R__30ped_grant);
 link_own_flow_to(R__30ped_grant, C1.ped_grant);
 link_out_flow(C2.b_red, b_red);
 link_out_flow(C2.b_amber, b_amber);
 link_out_flow(C2.b_green, b_green);
 link_out_flow(C1.ped_wait, ped_wait);
 link_own_flow_from(C1.ped_request, R__56ped_request);
 link_own_flow_to(R__56ped_request, C2.ped_request);
 link_out_flow(C2.ped_green, ped_green);
...

/* xtlight1.c – control_traffic_lights () – 1997-06-17 13:07:10 */
```

Figure 23. Piece of code generated by Reagenix.

PART IV: STRATEGIC COMPETENCE MANAGEMENT

This closing part of the dissertation focuses on how competence could, in practice, be managed in the context of the project-based relationships of a contract research organisation. The competence evolution framework that I have designed is an analytical model used by a researcher for understanding and explaining a complex phenomenon, not a managerial tool. The amount of empirical work required for analysing the competence evolution in the two cases was considerable. I will thus address only the first steps towards competence management in project-based relationships. However, without the empirical work even these steps would have been difficult to take – as opposed to what I had believed when starting my research.

9 MANAGERIAL VIEW TO COMPETENCE

In order to get a feeling of the differences between a manager's practical view and a researcher's analytical view to competence, Frame 4 in Section 2.1 and Figure 17 in Section 6.4 can be compared. The former was created by the author in an early phase of this research to illustrate the "actual project relationship and competence concepts" at VTT. As seen from the frame, the concepts were organised into three levels: "research and development", "management" and "strategy". The management concepts include project offer, project proposal, contract, project plan, project management group, project preparation and marketing task, project contracting and re-contracting task, project planning and re-planning task, project management task, project management group meeting, as well as managerial plans and reports. Organisational marketing plans, annual plans, and strategic plans are given as the main strategic concepts. Intra-organisational structures, planning and co-ordination tasks, and documents are thus emphasised as managerial tools at VTT. The case data that I have analysed and my personal experiences as a manager indicate the same. The approach could be called *intra-organisational competence management*.

The competence evolution framework summarised in Figure 17 includes quite different concepts: macro forces of the outer context; the inner context consisting of groups of actors having different objectives, goals and logic of action; and of the planned and realised project nets, in which mutuality can be assured and balanced; customer, supplier and co-operation processes; activities on resources to assure capability and balance particularity – that form the content of the logic of action – and the competence which results from these activities. Groups of actors, project nets, processes, process phases, resources, activities, and competence are then further classified. These concepts were used to make explicit and to analyse competence evolution over time. Based on the results of the analysis, an alternative managerial model, *relationship-based competence management* for contractual R&D, will presented and illustrated in the following.

9.1 INTRODUCTION

It is first of all important to realise that relationship-based competence management denotes focusing on inter-organisational relationships in the creation and use of competence. Since in this research projects are considered the most important relationship type of the focal R&D supplier, the management of competence is based on projects carried out with external parties. Referring to Figure 1, one could claim that this vehicle has already been heavily used. I agree only partially with this view. Instead of focusing on individual projects, it should be able to manage the whole portfolio of projects related to a specific competence area, be the projects green, blue or red.

As has already been mentioned, competence creation has revolved at VTT around green projects. They have been organised at ELE into so-called *strategic research programs*, involving associated blue and red projects. Each program has a steering group, including customers and partners interested in the subject of the program. Companies involved in these groups may provide case examples to be studied or prepared for subsequent blue and red projects. However, customers do not have much control over green projects, because these are funded almost entirely by VTT. The associated blue and red projects provide for the further development and leverage of competence, and it is encouraged to start preparing them as early as during ongoing strategic research programs. Figure 24 shows as an example the part of the strategic research program on embedded systems design (ELESJS), which was carried out to renew the fault diagnosis competence created and developed earlier in the nineties.

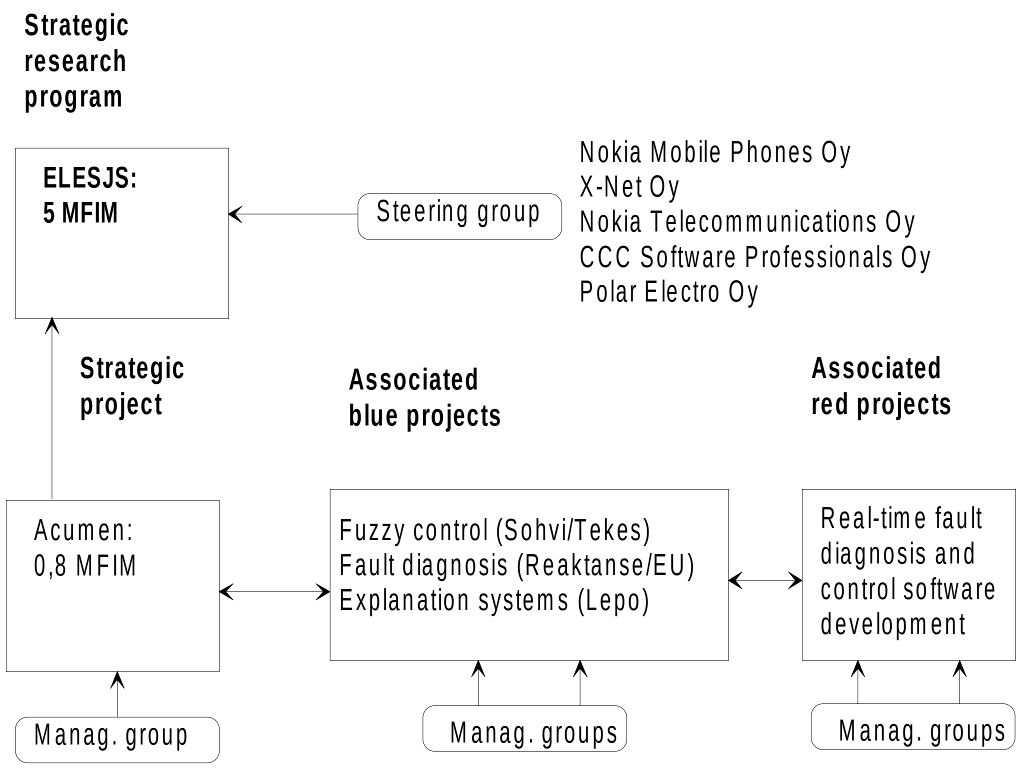

Figure 24. Part of the ELESJS research program in 1997.

The figure describes the situation in 1997. A new green project "Acumen" was launched that year involving visualisation and intelligent navigation of data – to re-direct the fault diagnosis R&D. The Acumen project was associated with three blue projects and several red ones. The five companies shown in the upper right corner of the figure were interested in the results and thus took part in the steering group. One of these companies financed a red project associated with Acumen. An attempt was made to launch a new blue Tekes-funded project in 1998, but this endeavour failed due to the lack of industrial interest. However, the associated red project was continued as a larger product development effort. In this project, VTT gained experience from new technologies and a particular application concerning real-time data logging and measurement. Furthermore, a formal agreement was written with the customer to ensure the continuity of collaboration.

The example shows one of the basic problems of the current approach: the ideal technology-push principle can not be guaranteed to work as intended. The code generation and fault diagnosis cases have been indicating the very same issue during the entire period of my study. From the viewpoint of relationships, this means that the transformation of intra-organisationally established and controlled green research activities into jointly co-ordinated blue networks, and further into several red relationships contracted by individual customers, may not be linear or it may even not be realised at all. The most obvious reason for this is also visible in Figure 24. The evolution of competence includes several alternative routes, which cannot be forced to converge during the initial research phases.

Although strategic green research projects may include associated blue and red projects, the allocation of resources to projects is determined by low-level organisational decisions. The decision of not allocating the key customer's former code generation expert to the Sokrates project is a very good example of this problem in the case data. In more general terms, market opportunities determine the launching of red projects. Blue projects are carried out in joint national and international research programs, which are funded by Tekes and EU, for example. Although it would be possible for VTT to affect these programs, it cannot direct them solely on the basis of the goals and schedules of its self-funded research.

Therefore, there should be a balance in the project portfolio as a whole, concerning not the ratio of self-funded, jointly financed and fully contractual projects, but also regarding the competence factor. Leppälä [1995] actually points this out by stating that it "is not sufficient in itself to balance contract R&D with strategic research projects. Organisational means should be fine tuned to accumulate core competence in the course of all research contracts." In other words, the line organisation and the project organisation are separate issues, and the *project portfolio management organisation* is missing at present. The lack of organisational support for systematic development of fault diagnosis competence illustrates the latter. The core platform evolved mostly aside red customer projects. In the code generation case, the Reagenix generator, which can be considered a renewal of the ideas experimented in Sokrates, was produced only after this project, i.e. the biggest blue code generation research project, had ended.

9.2 ANALYSIS OF MANAGERIAL IMPLICATIONS

Referring to Parts II and III, the key managerial implications of my two case studies will first be analysed and contrasted with the literature.

Context – economy vs. technology and organisations vs. individuals

- Changes at different levels affect competence evolution simultaneously.

The intertwining of inner and outer contexts makes individuals, organisations, inter-organisational relationships, industries, as well as national and global economies affect competence evolution. These phenomena are difficult, if not impossible, to predict, as is shown by both of my research cases. As an example, if it were known that expert system technologies would not result in any remarkable change in computing applications, not only individual researchers but also companies, industries and national funding bodies would have changed or cancelled their plans to investigate intelligent systems. Indeed, van de Ven et al. [1999] characterise the "innovation journey" as a non-linear and dynamic rather than a linear and predictable process – but not entirely chaotic and random either.

Actors – R&D parties

- Competence evolution proceeds through collaboration and conflicts.

The early phases of the code generation case illustrate that if the goals of different groups of actors are well aligned or if the resourcewise most powerful group dominates, competence evolves rapidly. However, conflicts also tend to lead to entirely new insights and are often the very source of new inventions. The above-mentioned fall of expert system technologies in the fault diagnosis case was a crisis that helped VTT to break free from these technologies, as well as from relationships that may have resulted in competition with the former collaborators. Quoting Bijker and Law [1992], "technologies and technological practices – are built in a process of social construction and negotiation", where they are often "born out of conflict, difference, or resistance". Resistance towards the dominating group of actors may also keep competence alive during difficult times, as is illustrated by the code generation case.

- Types of collaborators and interaction affect competence evolution.

The two cases show that exploiter or developer types of customers and focused buying or broad co-operating forms of interaction affect competence evolution differently, because customers' goals and involvement are different. However, there are no simple patterns to follow. For example, the exploiter type customers in the automation sector did not advance the code generation competence much, but were crucial for the development of the fault diagnosis platform. In the former case, the customers included both focused buyers (e.g., the firm Kc in Table 16) and broad co-operators (e.g., the firm Nm in Table 16). In the latter case, many of the initial exploiters changed into active developers when the "development" component of fault diagnosis R&D became important (cf. [Rosenberg 1976], pp. 75–79).

Relationships – R&D projects

- Competence creation and development take place concurrently.

As was already pointed out above, the ideal project portfolio from green to blue and red projects does not always work – since innovative R&D is a non-linear and dynamic process, involving many alternative routes with dead-ends and new beginnings, consisting not only of original inventions but also of improvements of existing solutions. The process-related part of the competence evolution framework (cf. Figure 14) assumes the linear competence evolution sequence. Although this was quite appropriate for analysing the two cases, i.e. to investigate activities on resources within the context of three different types of R&D projects, the same cannot be assumed for relationship-based competence management.

From the viewpoint of relationships, non-linearity can be understood as a consequence of the change of well co-ordinated internal activities – or dyadic relationships if the developments start from external rather than internal activities – to a very complex project marketing and purchasing horizon. Van de Ven et al. [1999] characterise this change as "the development and institutionalization of interorganizational relationships" with "high web dependence". In other words, the projects in a network of relationships affect each other in a complicated manner, as opposed to the view that they could be managed separately from each other.

Håkansson and Waluszewski [1999] discuss the effects of this kind of "path dependence" on investments made in resources, which is especially relevant to the resource-based view to competence taken in this research. As the authors point out, there are both considerable "heaviness" and "variety" to be found in resource collections, be they physical facilities or intangible knowledge. In other words, projects in which interrelated resources are created or used may either restrict or facilitate competence evolution in each other. Apparently, the former was true for many code generation projects and the latter common in fault diagnosis projects – where even the code generator was successfully used at some point.

- Without contractual projects competence is likely to wither.

Another way of stating this implication is to say that external exploitation is necessary for competence evolution. Rosenberg [1976] formulates the same phenomenon from the perspective of the users of novel techniques, claiming that "the development of human skills upon which the use of the new technique depends in order to be effectively exploited" is crucial. The fall of industrial interest in expert system technologies caused VTT to apply more established automation system technologies. This helped the development and exploitation of the model-based fault diagnosis approach, which was to become the foundation for the coming core platform. In the first GSM network diagnosis project the difficulties experienced in utilising the CBR technique that was new to the customer and VTT caused considerable problems. As illustrated by the code generation case, there is also the fallacy on intra-organisational exploiters – customers, who require free service.

- There are convergent and divergent periods in competence evolution.

Although the competence evolution framework was built for the purpose of understanding and explaining the changes in competence, it does not prescribe any ordering of changes. However, certain periods were obvious in both of the cases that I studied. They could be justified on the basis of the major changes taking place in project relationships, and offered useful milestones for the analysis of competence. For example, the third part of this dissertation shows that the period from 1988 to 1991 (called Sokrates) was quite different from the viewpoint of code generation related projects, when compared to the period from 1985 to 1987 (called Speco). As a result, competence evolved in a different manner during the successive periods.

Van de Ven et al. [1999] (page 185) present a cyclical model of divergent and convergent behaviours in the evolution of innovations. In the divergent phase organisations are "building relationships and porous networks", in the convergent phase they are "executing relationships in established networks". The former was prevailing from 1995 to 1997, when new application areas were sought for the fault diagnosis competence (cf. Table 30). The Sokrates period is a good example of the latter in the code generation case. From the managerial perspective, it would be important not only to understand and to make use of the prevailing period, but also to be well prepared for changes between the convergent and divergent periods.

Thomas et al. [1999] suggest – based on Hamel and Prahalad [1994], that the competition for the future opportunities of a firm proceeds in three stages. During the first stage, focus is on creating industry foresight and intellectual leadership, so that "firms will compete to develop a greater understanding of the technological, demographic, and lifestyle trends". This is clearly quite a divergent period. The main goal of the second stage is shortening migration paths, by "accumulating necessary competencies, testing out alternative product and service configurations, and attracting coalition partners with the necessary complementary resources and skills".

This characterisation resembles divergent times, as described by van de Ven et al. [1999]. Thomas and the others suggest that collaboration is important especially during this stage, when the limits of new technologies are explored and competence accumulated. The third stage, when firms look forward to ensuring market position and share, can be considered convergent, because "questions of technological platforms, rival product and service concepts, and distribution channels have largely been resolved", and external relationships are less important.

Activities on resources – carrying out R&D tasks

- Competence creation and reuse affect evolution.

If projects in which new competence is created and existing competence is reused and developed take place concurrently, the proportion of competence creation and reuse cannot be known by calculating the ratio of green, blue and red projects. Moreover, it cannot be managed by comparing the actual ratio to the expected distribution of projects in the ideal portfolio.

Figure 12 in Section 5.6, based on the managerial implications gained from the fault diagnosis case, suggests four types of situations regarding competence creation and reuse: breakthrough R&D (high competence creation rate, low reuse rate), incremental technology improvement R&D (low creation and reuse rates), needs-driven R&D (low creation rate, high reuse rate), and mass-customised R&D (high creation and reuse rates).

Ford et al. [1998] can be used to elaborate this view further, depending on the newness or familiarity of either applications or technologies as the key dimensions of competence content. If the technology is known and the application is unclear, "relationships are important" to the focal organisation, which will try to "seek new applications for known technologies".

Relationships are also important, when the technology is unknown but the application is clear, and when the goal is to "seek new technological solutions for known applications". In the case of a known technology and a clear application "development is ordered" and "single actors are important" – as also when the technology is unknown and the application is unclear, when "development is random" rather than ordered.

In other words, the authors consider the management of relationships important, when either the application or the technology is new. Most of the code generation projects analysed in this research represent the latter, i.e. there was a high creation rate of new technologies, whereas the knowledge of automation applications possessed by some of the focal code generation researchers dating back to as early as the seventies was reused. Aspects of creating new application-related competence can be found in the projects where the code generator was used to design integrated circuits (the Sasic project, Table 16) and to produce software for a novel scientific measurement instrument (the Kaasu project, Table 16).

In the fault diagnosis case, both the collaboration with automation companies in the late eighties and the first GSM network diagnosis project represent the novelty of technology as a motivation for establishing relationships – and thereby also the focus of competence. Most of the fault diagnosis projects carried out in the nineties for automation companies resulted from the need to develop fault diagnosis functions, and the knowledge of these functions could be reused by VTT.

The first GSM network diagnosis project illustrates that existing functions present a great challenge to the R&D supplier. The supplier's knowledge of new techniques and technologies usually helps to initiate projects, but the usefulness of this knowledge must be quite soon demonstrated with the customer's application. The need for new functions helps in this task. One of the very basic managerial conclusions from this research is indeed that it pays off to keep an eye on *functions* and *applications*, when trying to produce reusable while also extendable competence. This implication, which is well in line with Elfring and Baven [1996] discussed in Section 3.1, can be contrasted with the above-mentioned view of Ford et al. [1998] that emphasises the role of new technologies and applications as a reason for establishing relationships.

- Organisations and processes affect the evolution of competence.

In an engineering R&D organisational skills and processes provide support for technical competence, but cannot replace it, since the services that are being offered are based largely on technical capabilities. However, the lack of organisational skills and appropriate processes can seriously hinder the creation and development of technical competence, as is clearly shown by the code generation case.

Organisational competence has been associated quite straightforwardly with professional reputation in the competence evolution framework. From a managerial perspective, this is a much too limited view, since what matters in the end is how effectively competence is institutionalised.

The process-related type of competence concerns especially the management of R&D projects, which has been considered one of the strongest skills of VTT (cf. [Leppälä 1995]). However, what was said earlier in this section indicates that it is not obvious how the project portfolio is used, in practice, for systematically creating, reusing and maintaining competence.

9.3 RELATIONSHIP-BASED COMPETENCE MANAGEMENT

In the following, a model for relationship-based competence management in contract R&D will be presented. Referring to the managerial implications discussed above, the following aspects will be included in the model:

- changes of diverging and converging periods in competence evolution,
- separation between different types of customer organisations,
  ⇒ *Project portfolio design*: types of customers vs. types of services,

- interaction of relationship parties,
- coping with collaboration and conflicts between the parties,
  ⇒ *for customer focus:* level of mutuality and form of interaction,

- continuous interleaving of competence creation and development,
- emergence of competence due to its external exploitation,
- familiarity or newness of competence to the interacting parties,
  ⇒ *Concurrent R&D*: non-linearity of research, development and usage,

- interplay of technical, process-related and organisational competence,
- management of the content of technical competence,
  ⇒ *for core platforms:* management of particularity and capability.

Concerning these requirements, a project portfolio designed for achieving a specific customer focus ensures the supplier's position in the changing web of inter-organisational relationships, i.e. it constitutes the *relationship strategy*. On the other hand, making use of concurrent R&D to build core platforms can be considered a *competence strategy*, which is used for focusing on key resources and tasks, and which serves as a basis for creating and reusing the competence of the focal supplier.

9.3.1 Portfolio design for maintaining customer focus

As discussed above and shown by the case data of this research, there are diverging and converging times in competence evolution. This should be considered one of the basic challenges in designing the project portfolio. In the ideal portfolio shown in Figure 1, the *convergence of competence* from green to blue projects is, on one hand, assumed. On the other hand, the *divergence of customer relationships* can be seen as an implicit strategic goal, because investments made in green and blue projects should result in fully contractual red projects that provide for income and some revenue.

No explicit requirements can be set for the number of continuing red projects to cover the institution's own funding of green and blue projects. However, as was pointed out earlier, the national mission of VTT is to transfer new technologies to a large number of industrial customers rather than few. In principle, the industrial participants of green and blue projects, in which new techniques and technologies are often investigated, serve the goal of convergence regarding the competence and divergence of customer relationships: few triggering green dyads develop to many red dyads through a number of blue networks.

In practice, several of the industrial members of green and blue projects are interested only in early technology screening and testing. Thus, customers which are mostly or even entirely new need to be acquired for the red projects, where the results are intended to be further converged. The situation after the Sokrates project shows what may happen when a secure blue network collapses. Therefore, instead of aiming at a successive process of competence convergence and relationship divergence, these phenomena should be managed simultaneously. The goal must be to prevent the intertwined competence and relationship evolution process from stopping, which was one of the basic success factors of the fault diagnosis case.

Referring to the code generation case, both the *type of the customer* and the *form of interaction* are important for managing the relationship dynamics. For example, the firm Nm wished to exploit research results as a broad collaborator, whereas the firm N was interested in an operating system, a single result of the Sokrates project. In the current project portfolio, financing dictates the form of interaction between VTT and its customers. In green projects VTT dominates due to its own funding, in blue projects it often dominates because of its technical skills, but red projects are nearly always dominated by the customer, since it is paying for everything. This process presupposes radical changes in the interaction between the parties and easily results into conflicts – cf. the relationship between VTT and the firm Is regarding code generation technologies.

In the competence evolution framework, customers are divided into *developers* and *exploiters*, based on their view of competence. The competitive, collaborative and dominating/submissive forms of interaction are proposed. These become visible in the form of *focused buying* and *broad co-operation*. The former represents the most common dominating customer – submissive supplier situation and the latter the usual collaborative situation. However, both the specific customer type and the form of its interaction with the R&D supplier may change over time.

The readiness for collaboration and conflict resolution is modelled in the competence evolution framework by the concept *mutuality*. The functions of assuring and balancing mutuality are used for analysing mutuality changes. From the managerial viewpoint, mutuality is affected by the relation between the type of the customer and the form of interaction. With focused developer type buyers, mutuality is often at its lowest. at its lowest. The supplier is selling specific solutions to customers, who are or will be capable of developing and making use of the solutions by themselves. The above-mentioned development of the Sokrates-based operating system for the firm N (the Osdyn project in Table 16) is an obvious example. The change of the former exploiter type automation customers to active developers of fault diagnosis systems by the end of the nineties also represents rather low mutuality. None of the former key customers (cf. the examples discussed in [Kurki 1995]) have become long-term collaborators of VTT. The firms just needed external experts for some time, because they did not initially have the resources needed for developing fault diagnosis systems.

Long-term collaborative relationships are possible with broad co-operators, who are exploiters, provided that the R&D party becomes a system supplier. In the case of automation firms this would have required fault diagnosis systems to have become external modules of the customers' products. Instead, they became core parts of the products. With developer type customers, long-term relationships require strategic alliances, where competence may be diverging, but the relationship is converging.

Putting all these elements together, Figure 25 shows a project portfolio by which the supplier can manage its customer focus and thereby the evolution of its competence. The management of divergent and convergent periods relies on employing the conceptual framework consisting of customer types and forms of interaction; the changes these may experience are also considered, with increasing (+, ++) or decreasing (–, – –) mutuality between the relationship parties.

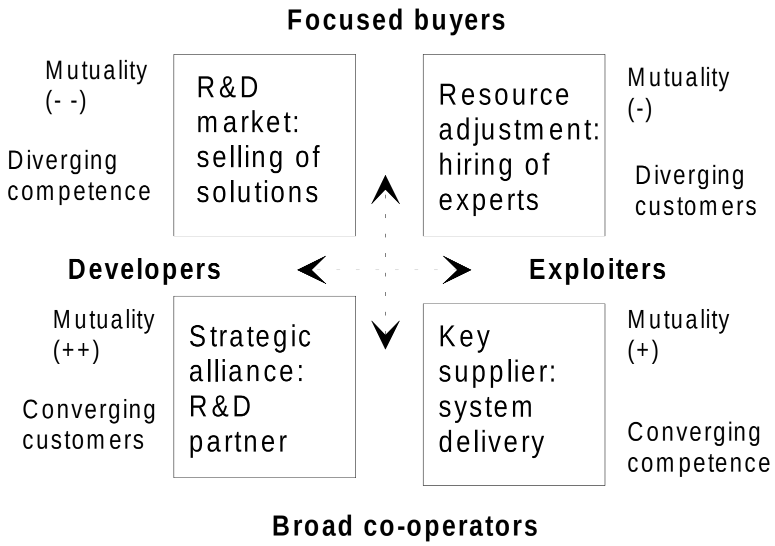

Figure 25. Project portfolio design for maintaining customer focus.

The relationship strategy of a supplier would require identifying the current and target portions of the project portfolio of a certain R&D topic, or choosing one to start from, if the topic is new. This would affect the competence strategy discussed in the next section. In the case of VTT, green, blue and red projects would still form an important financial framework for projects, but not the basis of strategic relationship management. Each of the four portions of the portfolio shown in Figure 25 could include a combination of green, blue and red projects. In other words, customer relationship and competence evolution would be managed by adjusting mutuality, not on the basis of project financing. The four portions of the portfolio characterised in Figure 25 as *R&D market, resource adjustment, strategic alliance* and *key supplier*, will be discussed below. The portions are, in practice, not clear-cut, and a customer can change its position due to intra-organisational or external reasons. For example, the firm K has developed from a developer type broad co-operator into an exploiter type focused buyer in code generation projects.

Focused buyers – diverging times

As shown in Figure 25, the R&D market consists of focused developer type customers, whereas focused exploiter type customers are interested in resource adjustment. The two situations imply divergence in competence and customer relationships, correspondingly. Another way of describing the upper portion of the figure is that the supplier needs to market specific elements of its existing competence to new customers and be prepared for creating new competence for the needs of existing or new customers. In the former case, the expected length of a customer relationship is short, in the latter it is difficult to maintain an adequate level of competence.

R&D market – how to prevent competence from fragmenting?

Mutuality is low in the R&D market, since the supplier is selling specific solutions to customers, who will be capable of developing the solutions themselves sooner or later. If the supplier establishes too close ties with the customer, the parties may become competitors. The earliest fault diagnosis projects carried out in the eighties can be considered an example of the R&D market situation, although VTT apparently considered them a strategic alliance, because mutuality was perceived as high.

The reason is that the customers' knowledge engineers with whom VTT was developing prototypes of expert systems were focused buyers, looking for specific skills, and aiming at utilising the skills by themselves for the needs of their own intra-organisational customers. In the code generation case the Nosto project (Table 16) provide an even better example of the R&D market situation. The firm Kc, for which an application-specific code generation environment was developed by VTT in 1993–1996, has not been involved in any other projects afterwards. Competence was diverging in these projects, since distinct customers were seeking for particular solutions tailored for their needs. Even if the supplier would produce uniform and generic solutions, such as the solution elements developed by the code generation researchers during the Sokrates project, they would need to be sold to many customers.

Customer projects are likely to be sporadic. In other words, there will be quite an number of different customers. The breadth of the project portfolio increases, but only a few continuing projects may realise. The reason is that customers seek for specific skills that they are likely to learn themselves quickly, if they wish to start using them as part of their own development activities. These skills are most often technical, involving new technologies and techniques – such as the expert systems technologies and knowledge engineering techniques in the late eighties, and the structured system design techniques and the code generation technologies at the turn of the nineties.

The R&D supplier should make use of the divergent times, though, instead of trying to resist them desperately. Collaboration with the knowledge engineers of automation companies interested in expert systems was quite useful for VTT, and without the industrial interest in code generation the Sokrates project, for instance, could never have been established. Moreover, the code generation researchers encountered difficulties due to their reluctance to make use of the rapidly emerging object-oriented techniques instead of the structured techniques that were withering.

Blue projects are a typical example of the R&D market in the case of VTT. Universities and research institutes look for industrial parties, and industry is interested in learning about new technical solutions. However, the ideal project portfolio shown in Figure 1 assumes that competence is converging, not diverging, in blue projects. If this happens, e.g. in such a way that the results of a blue project are developed in close collaboration with an industrial partner, it may easily happen that the supplier becomes misused. It gives away its new skills without much continuity in the customer relationships, other than perhaps one red project in which the competence is institutionalised by the customer. Not even a single continuation project was established with the firm Is regarding the code generation technologies, quite contrary to the expectations of VTT. In the worst case, a developer type customer interested in focused buying is taken as a strategic partner in a green project. However, no such developments have taken place in the two cases studied of this research – there were no industrial parties directly involved in the self-funded research projects carried out in the nineties.

The benefits of the R&D market arise from the possibility of creating and renewing competence for particular customer applications. This presents a managerial challenge especially in red projects, but also in blue projects where some concepts need to be tested for varying needs. The basic conflict is that the external party pays for and wishes to obtain particular solutions, whereas the supplier would need to maintain its own competence. In the fault diagnosis case, this has succeeded in the nineties thanks to the core platform concept, which allowed the fault diagnosis functions and the process of developing fault diagnosis systems to be reused, while applications, technologies and techniques would vary according to customer requirements. An important managerial question, when facing the R&D market, is thus how to establish projects that help to create or renew competence, but do not result in the fragmentation of competence due to varying customer needs.

Resource adjustment – how to institutionalise competence?

The resource adjustment situation means that exploiter type focused buyers make use of an external supplier for conducting some tasks that they cannot carry out themselves for the moment, because they do not possess the necessary resources. As discussed above, exploiters may wish to become developers. This is rather typical among the customers of VTT, most of the fault diagnosis projects represent this case. A good example of a similar situation regarding code generation is the firm K, although it ultimately turned into an exploiter and bought a code generator from the firm Is.

In terms of project portfolio, resource adjustment would often denote red projects being carried out for individual customers – or blue projects involving several industrial participants interested in test usage of the developed solutions. In both cases, the customer is interested in making use of certain competence, which means that there is some mutuality. Figure 25 indicates that here mutuality is higher than in the case of the R&D market consisting of developer type focused buyers, which may be surprising. However, the results of this research indicate that it is not the new technology or technique as such that is important, but its use by someone.

Developer type customers tend to restrict the supplier's access to the end-users of the developed solutions, be the users internal or external. For example, no telecommunication operators were involved in the first GSM network diagnosis project carried out by VTT. When a project was then established with an operator, no network equipment manufacturers were involved. The early fault diagnosis projects are an exception. However, if the solutions had been taken in industrial use, VTT would most likely not have been involved in the continuing projects, but only the customers' R&D laboratories as the supplier and business units as end-users.

Mutuality between exploiter type focused buyers and R&D suppliers often implies, as is indicated in Figure 25 ("experts"), personal rather than organisational relationships. For example, the relationships between the VTT researchers and the customer's engineers in the MCS-REA project (Table 16) were quite close indeed. From the viewpoint of competence evolution, this may result in problems regarding the institutionalisation of the competence, since individuals possess customer-specific skills. In the fault diagnosis case, such skills were institutionalised to portfolio-specific competence in the early nineties, by focusing on automation applications.

This is the basic managerial question in resource adjustment. In other words, competence can be converged, if the diverging customer relationships show any focal areas. Without such areas the competence will be marketed to many different exploiters, and it will start diverging. The attempt to apply the Sokrates code generator in the design of ASIC circuits would have resulted in such divergence, if it had succeeded. Instead of the focal code generation researchers, the ASIC researchers were responsible for the divergence. In the fault diagnosis case, the evolution and extension of the core platform was done by focusing on automation applications.

Broad co-operators – converging interests

The possibilities of the R&D supplier to become a long-term collaborator with a focused buyer depend on the customer's strategic goals regarding competence. Developer type customers usually focus on their own core competence. Although the supplier may be used to renewing parts of this competence, the customer does not usually wish to become too dependent on external suppliers. Longer-term relationships may emerge if, for example, the renewal rate of the customer's competence is rapid – as is often the case in the electronics industry. The supplier may misinterpret this as a strategic partnership, although Figure 25 indicates that a broader basis is needed for the collaboration to turn into strategic alliance.

A developer may, however, become an exploiter regarding a specific competence, which provides new opportunities for former suppliers. This kind of change concerning the firm N in the early nineties would not only have prevented the code generation competence from withering at VTT, but it would also have resulted in extraordinary possibilities. This is what happened to a software testing tool in the nineties. The code generation researchers, however, criticised the development and upgrading of this tool. They regarded it as misuse of customer relationships.

Key supplier – how to avoid core rigidity of outsourced systems?

Whereas the upper part of Figure 25 implies less mutuality and more changes in competence and relationships of the R&D supplier, the lower part indicates higher mutuality, broader collaboration and less divergent times. Another side of the coin is the need to invest in long-term customer relationships, because mutuality must be realised. The figure suggests that with exploiters this often results in a situation where the supplier takes responsibility for a particular system on behalf of the customer.

Externally acquired parts of products are common in industry. However, for such research institutions as VTT it is not easy to become a key supplier for a specific customer. This may succeed if the required competence involves continuous research and development. This is the case with the software testing tool mentioned above. VTT has taken an organisational responsibility for the further development – and even the usage support of the tool. In principle, such relationships could have been developed with the firms K and Nm regarding code generation. However, the former opted for a commercial tool vendor and the latter gave up using code generation.

The fault diagnosis case data shows that many of the former exploiters have turned into developers. However, at least in one occasion the customer has made a contract with an external company regarding the further development of a part of its fault diagnosis system. Although being a key supplier does mean converging competence for the R&D supplier, it is important to prevent the relationship from becoming too well-established. This can easily lead to routine work done for a customer, who has reserved only few resources for the work itself, because it relies on the key supplier's services. In other words, a well-established key supplier type of relationship can lead to core rigidity [Leonard-Barton 1992].

Strategic alliance – how to stay as an R&D partner?

Developer type customers can establish strategic partnerships with selected R&D suppliers. This kind of relationship is common between corporate research centres and business units. This type of relationship requires a lot from an external partner, especially concerning the in-depth knowledge of the customer's application, and thus also a very high degree of mutuality. To become a partner may not be as difficult as it is to stay in that position, because the customer is continuously improving and extending its own skills. The supplier should be as good as the customer, or better, if the customer's and the supplier's competence are closely related to each other.

It is illustrative how VTT has neither become a strategic partner in code generation nor in fault diagnosis with any customers during the period from 1985 to 1997, which was investigated in this research. The firm N could have become one, but VTT missed the opportunity of making itself more deeply involved in the code generation research and development. An even more interesting question is why VTT failed to continue as a strategic partner of the automation firms with whom it was able to build the core fault diagnosis competence. There are several possible answers, which are related to the problem of maintaining a close collaboration partnership.

First, regarding competence, the customer may become an exploiter or a focused buyer. VTT actually started as a strategic partner of the firm K in code generation R&D, but the customer first turned into a focused buyer and then into an exploiter. The code generation researchers told in the interview that the managers of the firm K were pursuing the goal of becoming a focused buyer of the Sokrates code generation technologies, but that the customer's software designers opposed this idea. The firm's decision to become an exploiter was, in the end, due to a corporate-level re-focusing of its technology strategy. It can also be seen as a more general result of not only the recession, but also of the fact that the vision of embedded computers becoming a core technology in industrial automation systems was not realising as expected in the eighties.

Second, there is always the risk for the solutions developed for the customer becoming outdated or even obsolete. In other words, the techniques and technologies that have made the R&D supplier and customer collaborate, lose their role as a basis of the relationship. If VTT and the automation companies involved in the first fault diagnosis projects had established strategic partnerships concerning the development of fault diagnosis systems, the alliances would finally have collapsed due to the fall of the expert system technologies. The resulting obsolescence of the expensive system development tools acquired by the parties illustrate the dangers involved in the technical foundation of customer relationships.

Third, the customer may simply become dissatisfied with the supplier, as the firm Nm illustrates regarding code generation. Moreover, the customer may decide to start maintaining the competence by itself. The fault diagnosis projects carried out by VTT for automation companies in the nineties represent the latter, the customers did not "need" VTT any more. They had learnt how to build fault diagnosis systems by themselves.

9.3.2 Concurrent R&D for core platforms

Competence strategy consists of the processes and resources used by the R&D supplier for creating, maintaining and delivering its competence. These processes are non-linear and concurrent, and conducted within the context of the portfolio of relationships shown in Figure 25. As discussed above, concurrence means that the R&D process is neither linear nor chaotic. Actually, there is no single sequential R&D process (cf. the innovation, development, leverage and commercialisation phases of the R&D process shown in Figure 14), but *several parallel R&D processes*.

Concurrent and non-linear R&D processes

Since the goal of the supplier, based on the results of this research, should be to make its competence explicit and to evolve it towards core platforms when providing R&D services to customers, the two basic processes that need to be concurrently conducted are the *R&D service process* and the *platform management process*. The processes are associated with each other by two additional processes: the *platform reuse process* used in R&D services; and the *service reusability process,* by which reusable competence elements are extracted from the delivered R&D services.

These processes are visualised in Figure 26 by means of an X-shaped process framework. As shown in the figure, yet another concurrence results from several projects of the portfolio being carried out simultaneously. They can all contribute to the emerging competence, depending on the content of the service. The several concurrent platform management projects could also be represented in the figure, but they have been omitted for the sake of simplicity. The concurrence of the processes and projects forms the foundation for the non-linearity of R&D. For example, if a specific competence element is used with some modifications in several projects, be they green, blue or red, the modifications may very well involve research, development and leverage. This happened, for example, in the blue Tulko and Rulla projects (Table 16) with the code generation competence. In the ideal process only development would have been carried out.

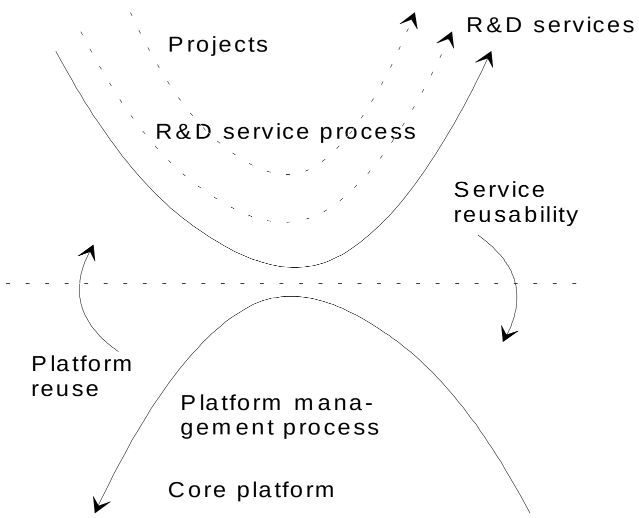

Figure 26. Concurrent R&D processes.

Reuse-based R&D services – managing diverging customer relationships

The R&D service process involves conducting the contract research and development so as to create and transfer resources to the customer. This is one of the key processes at VTT, and it has been discussed throughout the dissertation. However, as opposed to the stand-alone process depicted in Figure 14, the service should be based on the reuse of the core platform. Competence reuse facilitates managing diverging customer relationships, as is illustrated by the fault diagnosis case. On the other hand, new relationships contribute to the renewal of existing competence.

Referring to Figure 25, competence reuse is expected to be rather low in the R&D market and resource adjustment cases, since focused buyers tend to seek for specific services. However, reusability can be achieved as such or by tailoring (cf. absorption in the competence evolution framework) existing competence to the customer's needs. For example, one of the goals of the product-based logic of action of the code generation researchers was to deliver generic, well-packaged competence elements to customers. In practice, rather extensive tailoring of the competence was required – for example for the firm Nm in the MCS-REA project (Table 16), and competence reuse was not as extensive as planned. The internal customers utilising the Reagenix code generator in the late nineties could, on the other hand, reuse it much as such and without the help of the focal researchers, although this represented a typical resource adjustment situation.

In the fault diagnosis case, the tailoring of competence involved the creation of models of the customers' applications and the implementation of the fault diagnosis systems using the technologies favoured by the customers. The tailorability of techniques was supported by maintaining a pool of generic knowledge engineering techniques. The fault diagnosis functions and the process for designing fault diagnosis systems could, however, be reused to a high degree as such. This prevented the competence from diverging, although there were many exploiter type customers. The high level of reusability of the competence in the nineties is also due to the fact that the projects were carried out for companies in a limited industrial sector. The applications did not diverge as much as, for example, in code generation.

Although the platform reuse process will not be treated in detail here, it should be noted that if specific competence elements incorporated in the core platform are reused with modifications, they must be selected (cf. scanning in the competence evolution framework) and internalised by the project team. If the elements are reused as such, they must be combined with other elements [Nonaka and Takeuchi 1995]. Consequently, the platform reuse process affects both the capability and particularity of the R&D supplier's competence. In case there are several projects in which competence elements are reused and integrated with other elements, the service reusability process becomes critical for the R&D supplier's ability, in view of strengthening and institutionalising its competence. Despite Despite its importance, this process has not yet been made explicit at VTT.

Platform management process – making competence converge

The platform management process, as described in Figure 26, is based on producing reusable competence elements from projects where R&D services are being conducted for customers. Referring to the relationship strategy shown in Figure 25, this emphasises the role of broad collaborators – be they developers or exploiters. Although it is possible to produce reusable competence elements in projects carried out for focused buyers, their strategic importance for the supplier may not be high. Good examples are the Sympa and Cute projects carried out for the firm S (Table 16), which took part in the Sokrates project. A software test tool was produced in these projects, based on the ideas investigated in Sokrates. The tool was marketed by VTT to many customers, but it remained isolated from other competence related to software testing and did not have too much success.

The service reusability process requires generalisation (cf. problem solving in the competence evolution framework) of the competence elements produced in distinct projects, and their integration into the platform. This means that the project team, with the help of the platform management organisation, selects elements from what they have learned and produced (cf. scanning in the competence evolution framework) in the project, and then externalise or combine [Nonaka and Takeuchi 1995] these to make them serve as parts of the platform. This process was implicit even in the fault diagnosis case, where the core platform emerged during the nineties. It could be carried out, though, because fault diagnosis was the key competence area of the focal research group. In other words, the fault diagnosis knowledge was both socialised and externalised within the group.

In the platform management process new competence elements are created and acquired, while existing elements are either removed from the platform or transformed into another form. The goal is to ensure the capability of the supplier in its future relationships. It is worth pointing out that VTT has usually not used subcontractors for creating new competence or maintaining existing competence. The case data of this research includes only one exception, the green Aniprosa project in which the firm El was a subcontractor, satisfying a need of the resource adjustment type (Table 16).

Using its own subcontracting relationships might have helped VTT to lengthen its collaboration with some fault diagnosis customers in the late nineties. The customers were looking for someone to maintain and to take care of the systems developed by VTT. If the platform management process had been made explicit, it is most likely that the use of external partners would have been realised, not only in contractual customer projects but also in the management of the fault diagnosis competence. Thus it would have been possible for VTT to assume a role as a system supplier. As pointed out in the cross-case analysis in Section 8.2, some customers decided to contract parts of the developed fault diagnosis systems to third parties. Using subcontractors could prevent not only routine work done for exploiters, but also expensive investments in special skills needed in strategic alliances with developers. As an example, the company established by the former code generation researchers was used by VTT for the latter in 1999.

Building of core platforms – the content matters

The relationship strategy affects the particularity of competence. Solution selling and system supply presuppose explicit codification of competence. In the case of solution selling, developer type customers are aiming at integrating the supplier's work and its results into their own development tasks and resources – consider the Osdyn project (Table 16) carried out for the firm N, for example. The resulting operating system became one of the most reused solutions of the customer's product. In the case of system supply, the exploiter type customers must be able to use the developed system effectively. This did not succeed with the code generation environment developed in the MCS-REA project for the firm Nm (Table 16).

Resource adjustment and strategic alliances tend to result in tacit or portfolio-specific competence. In resource adjustment, exploiter type customers are seeking for skills that they do not possess, and perhaps do not even wish to possess if their aim is not to become active developers regarding the specific skills involved. For example, VTT Food technology laboratory – like many of the other internal customers – was interested only in using the Reagenix code generator in the Kaasu project (Table 16). In strategic alliances, the close relationship often results in tacit knowledge shared by the parties. VTT became a close collaborator of the firm K in the mid-eighties, in part since several key persons moved from VTT to the customer company. The Speco projects, i.e. the first code generation related R&D activities, can be traced back to a shared understanding of the managers of the parties on the future of embedded software engineering.

As shown in Figure 27, capability is the another aspect of competence affected by the competence strategy. By increasing (+, ++) or decreasing (–, – –) capability and particularity in relationships, the strategy can be expressed through four competence areas, *scientific*, *problem-driven*, *technology-driven* and *application-oriented*. Moving from one area to another means focusing on two of the four dimensions of competence.

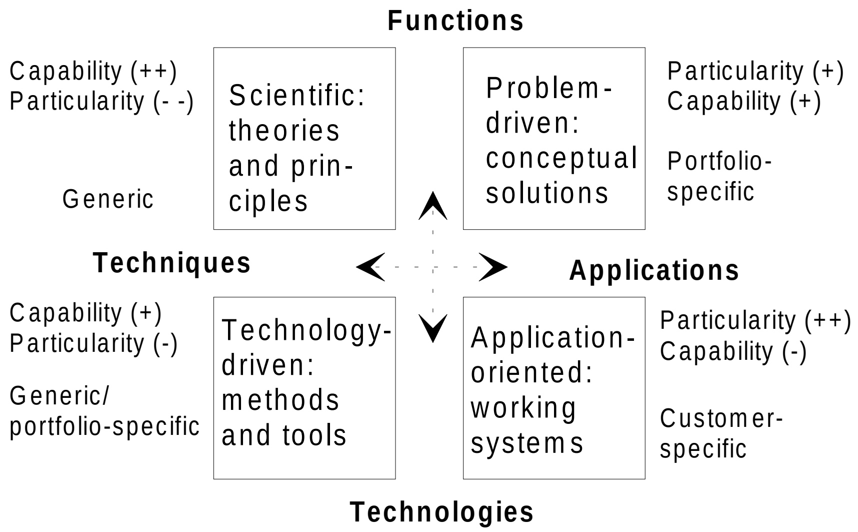

Figure 27. Competence strategy.

Scientific and technology-driven competence – focused buyers

The left-hand portion of Figure 27 indicates quite high capability, i.e. advancing the life cycle of competence, and rather low particularity, i.e. generic or at most portfolio-specific competence. If the R&D supplier emphasises techniques and functions, it is interested in creating new theories and problem-solving principles. These can be considered scientific competence. The code generation researchers' work of investigating and developing rules that made it possible to generate program code directly from design models provides a good illustration of this sort of scientific competence. The work started already in the Speco projects, and was continued even after the Sokrates project, during the development of the Reagenix code generator.

Although some VTT colleagues criticised the researchers' work, claiming that it did not represent the correct scientific research process, the results were inevitably novel – even to the extent that they were patented. This can be seen as a demonstration of the researchers' wish to radically change the life cycle of embedded systems design principles. Moreover, the design principles were intended to be generic, to allow use in any applications, rather than just in the field of embedded systems. Later, the code generation researchers turned into technology-driven competence of a kind, as the research results were packaged in the form of an explicit method, including Sokrates-SA, the code generators, and some system solutions. Capability decreased – consider the late replacement of the structured design techniques, whose dominance was weakening. However, the particularity of the code generation competence increased due to the focus on automation applications. Some parts of the competence became portfolio-specific.

All the code generation customers listed in Table 16, except for the firm Nm and VTT Electronics laboratory, were focused buyers. In particular, the industrial members of the Sokrates project were more interested in specific results than in software development processes and environments based on comprehensive code generation. In more general terms, scientific and technology-driven competence are typical of relationships with focused buyers, i.e. the R&D market and resource adjustment situations.

In the fault diagnosis case, scientific competence dominated during the late eighties, when knowledge-based expert systems were investigated for fault diagnosis by industrial knowledge engineers. Such systems had not yet been extensively built in Finland, and the principles of developing them needed to be found, for example, from the scientific papers written by famous foreign artificial intelligence researchers. Although the systems were built for particular customers, application-related knowledge was embedded into the so-called knowledge bases, whereas the systems were built by using fully generic knowledge engineering techniques and expert system technologies. The knowledge bases were thus the only particular elements of the systems. The attempt to use the new CBR technique in the first GSM network diagnosis project bears a resemblance to these early projects. As pointed out in Section 5.4, the technology-driven competence did not actually play any central role in fault diagnosis projects especially in the late nineties, when VTT was collaborating with various automation companies.

Problem-driven and application-oriented competence – broad co-operators

The right-hand portion of Figure 27 indicates a higher particularity and a somewhat lower capability in the supplier's competence than the left-hand portion. Moreover, since the competence is portfolio-specific or customer-specific, customer relationships are quite close. In terms of relationship strategy this implies broad collaborators. In the problem-driven case, the supplier aims at producing conceptual solutions based on its understanding of the functions needed in a particular portfolio of customer applications. If it focuses on technologies instead of functions, its competence tends to become mainly application-specific and its main goal is likely to be to build working systems for individual customers by means of the technologies.

One could assume that exploiter type customers would cause the supplier's competence to become application-specific and that developer type customers would help the supplier to create conceptual solutions. However, the fault diagnosis case indicates quite the reverse. The likely reason is that the automation companies for which fault diagnosis systems were developed during the nineties, were not keen on the latest technologies, but wanted to have fault diagnosis functions in their products. These functions were required by the customers of their own customers, i.e. the users of automated processes and products, and the functions were not yet available.

In the GSM network diagnosis project, VTT was aiming at a strategic alliance with a developer type customer. However, the related functions had already been implemented using the technologies known by the customer. These functions could not be replaced by VTT in the project, which had to content itself with merely demonstrating some new or improved functionality based on the CBR technique. In other words, the customer became a focused buyer and VTT was forced to try to sell a new solution. One of the two customer business units bought the idea of the solution for one of its future products, whose development was about to start. Consequently, VTT had to increase the particularity of its competence considerably, and could not much reuse the core platform developed in the projects carried out for automation customers.

Interestingly enough, the firm Nm, the most important broad co-operator in the code generation case, had originally been interested in utilising the code generation functions. This succeeded well in the Kaapeli project (Table 16) – no similar functionality had been available to the customer earlier. The conceptual solution was then implemented as a technological code generation environment in the MCS-REA project (Table 16), to be used by the customer for automating the software development for certain industrial systems. As an exploiter, the customer did not wish to get involved with the technical details of the environment.

It was claimed earlier in this dissertation by the author that it was a mistake to hide the functionality inside the code generator. In other words, it has now been demonstrated that problem-driven competence that results in conceptual solutions forms a good basis for core platforms. This requires the building of a portfolio of relationships with broad co-operators, be they developers or exploiters.

Summary

Figure 28 summarises the fault diagnosis case, using the relationship-based competence management model. In 1985–1991, there was an R&D market with a few focused developer type buyers, with whom VTT was aiming at increasing its competence in knowledge engineering techniques and expert system technologies. This changed in 1992–1994…1998 (arrow 1) into resource adjustment for exploiter type automation companies, and in 1995–1997 into a few key supplier relationships (arrow 2.2). The core fault diagnosis platform was built by VTT on the basis of problem-driven competence, focusing on functions and automation applications.

The first GSM network diagnosis project turned into an R&D market situation in 1995–1997 (arrow 2.1), in which the aim was to combine the scientific competence with new functions. At present, the endeavour may be turning into a strategic alliance (arrow 3) based on problem-driven competence, now that VTT has been accumulating its application-specific knowledge. The figure also shows that the focus of competence has changed from knowledge engineering techniques to applications (arrow a) and from expert system technologies to fault diagnosis functions (arrow b), and that the exploiter type automation customers have turned into active fault diagnosis system developers (arrow c).

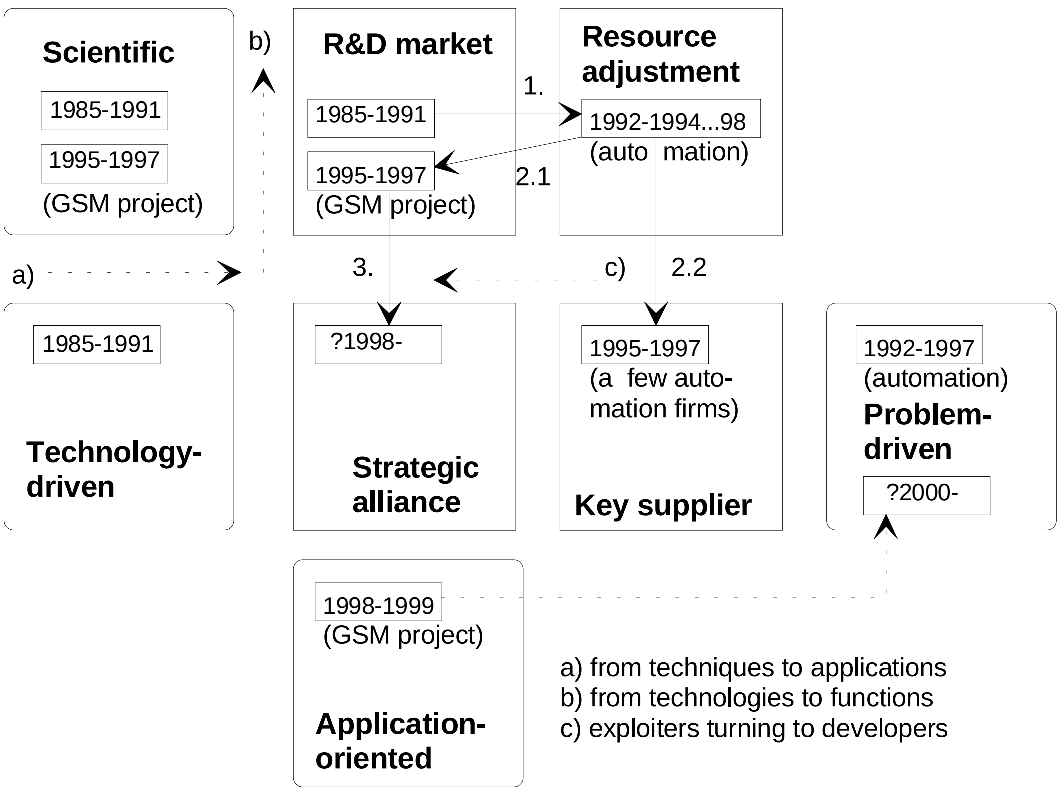

Figure 28. Relationship-based management of fault diagnosis competence.

Based on the figure, the initially diverging competence (R&D market) has changed into diverging customers (resource adjustment), i.e. to two groups of relationships, one with exploiter type automation firms and the other with a telecommunication firm. A few of the former have converged as VTT has became a key supplier, while the competence has converged into the core platform. However, when the customers turned into developers, the relationships did not converge into strategic alliances, but broke down. The relationship with the telecommunication firm still holds the promise of such an alliance, after having passed through the R&D market situation.

Figures 29 and 30 summarise, in contrast, the code generation case using the relationship-based competence management model. Figure 29 clearly shows the two important characteristics of the code generation projects: none of the relationships with developer type focused buyers resulted in strategic alliances (a reversed change took place with the firm K), and only one of the many relationships with exploiter type focused buyers resulted in a key supplier situation (the firm Nm). In other words, the code generation competence and the customer relationships have diverged considerably.

The roots of the divergence of the competence are in the Sokrates project – the most extensive code generation network in which more than ten companies participated. Ironically, one of the reasons why the project was considered particularly successful in the Finsoft research program was the large number of its industrial participants. However, most customers did not continue their relationship with VTT any further. The reason for only one key supplier arrangement having developed from the numerous projects of the resource adjustment type is that most of the projects – actually all but Kaapeli and Raski (Table 16) – involved internal customers, i.e. were carried out by VTT itself. As was discussed earlier in this dissertation, the participants of these projects did not show much interest in developing code generation competence any further, rather they were interested in using effective software tools for free. Moreover, the projects were not any long-term strategic partners as customers, but only temporary organisations.

The divergence of the code generation competence is presented in Figure 30. The originally scientific and technology-driven competence (cf. the fault diagnosis case, where the initial situation was much similar) would soon diverge into all possible directions. As shown in the figure, many of the projects involved different types of competence simultaneously, e.g. the Sokrates project included all four types of competence. The MCS-REA project, which was the most important key supplier arrangement, included problem-driven, technology-driven and application-oriented competence.

Although there were quite a number of application-oriented projects, these included mainly internal VTT customers which were not interested in viewing code generation from the problem-driven perspective. A clear exception is the Sasic project, were the code generation principles were developed further to satisfy the needs of ASIC design. The code generation principles, however, turned out not to be suitable for the purpose. More generally, the number of projects in which problem-driven code generation competence could have been developed, was very low.

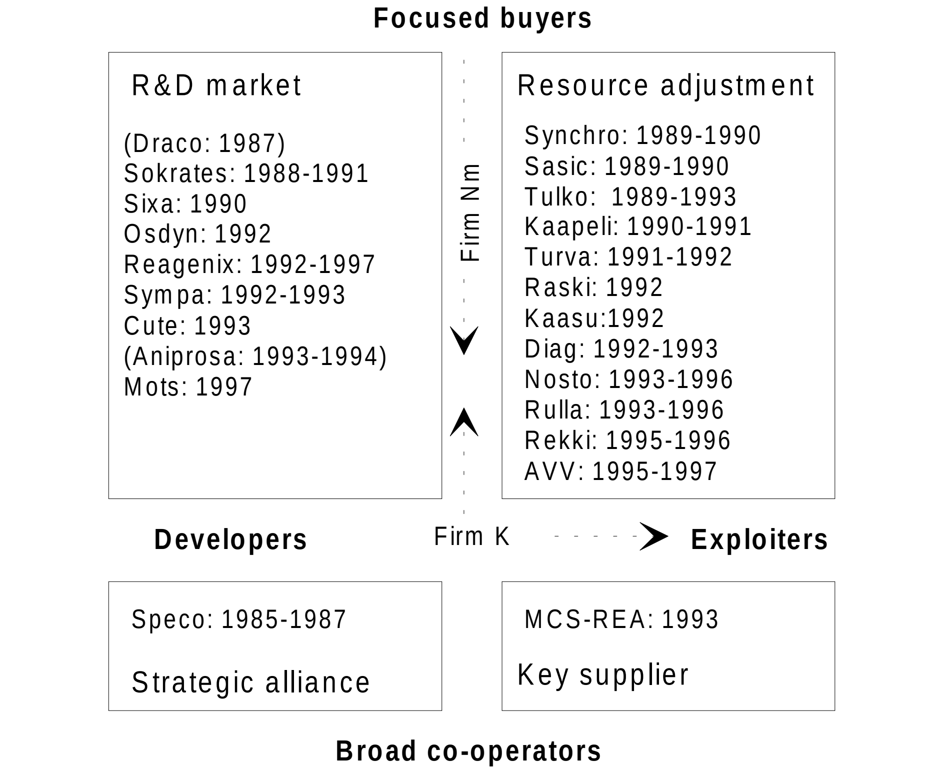

Figure 29. Portfolio of code generation projects 1985–1997.

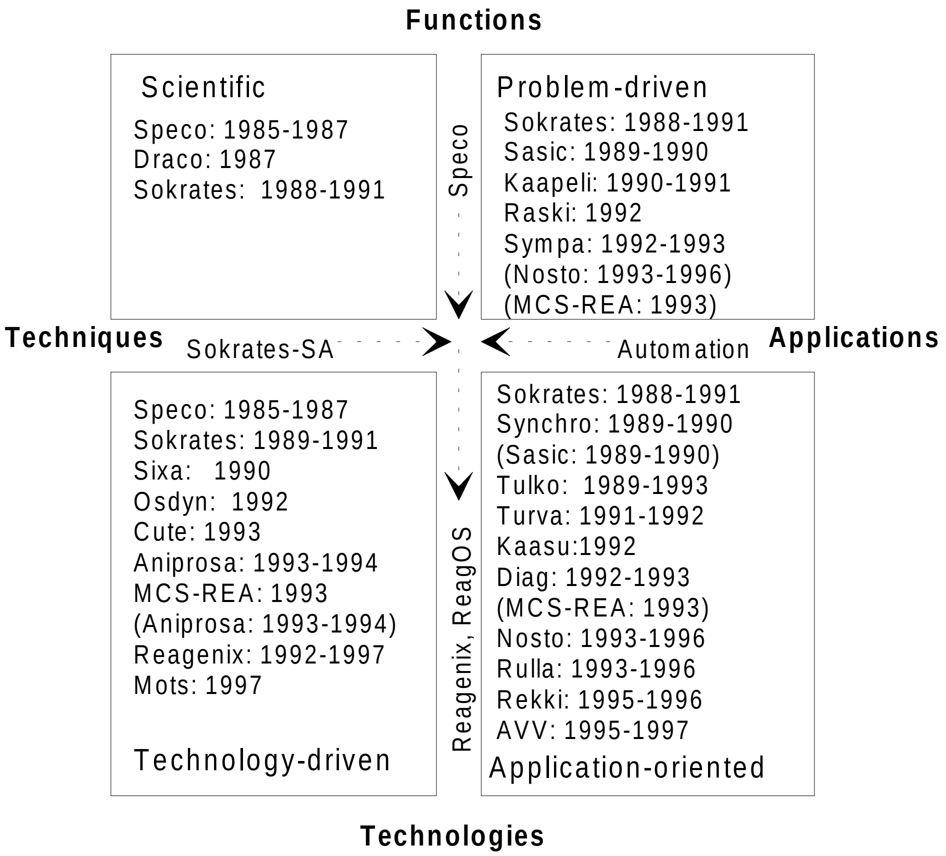

Figure 30. Code generation competence 1985–1997.

The relationship-based competence management model includes concurrent R&D processes (cf. Figure 26). Both of the two cases involve concurrency in the form of different projects being carried out simultaneously. However, there were no explicit service reuse, platform management and platform reuse processes – provided that Reagenix is not considered a core code generation platform. Referring to Figure 22, it cannot be regarded as such, as it remains a black box to anyone else but its developers.

9.4 DISCUSSION OF THE MODEL

The relationship-based competence management model will be discussed in the following, looking at the history of technology for points of comparison, since the model emphasises the change of relationships and competence over time. In view of the fact that the services offered by VTT as a contract research organisation involve not only economic but also social aspects due to close relationships with customers, I will also refer to research involving a social view of technology management.

Rosenberg [1976] has investigated, among others, the machine tool industry in the late twentieth century (pp. 9–31), which essentially contributed to the industrialisation of the Western economies. He points out that there was an increasing vertical disintegration of single industries, accompanied by "technological convergence of larger groups of industries", i.e. diffusion of new technical solutions from an application for which they were originally developed to entirely new applications. A similar process is represented by the fault diagnosis solutions developed by VTT for heavy work machines being carried over to GSM network equipment. The process is affected by existing solutions in the target industry, where individuals and organisations may consider that "[new solutions] may offer only small advantages, or perhaps none at all, over previously existing solutions" (p. 195).

Rosenberg also refers to single components or parts of complex systems and individual operations of comprehensive industrial processes as possible limiting factors for the change: "single improvements tend to create their own future problems, which compel further modification and revision". An excellent example of future problems created by single improvements is the decision to use the SA/SD method in code generation by VTT. On the other hand, "complementarity" in productive activity between different techniques" (p. 201) plays an important role in the diffusion and expansion of new technical solutions. For example, knowledge engineering and conventional control system design techniques co-existed in the model-based fault diagnosis approach (Figure 11), whereas Sokrates was aiming at entirely replacing the earlier embedded software design techniques.

Rosenberg speaks quite strongly against the separation between *inventions* and *innovations*, since "the productivity-increasing effects of technological change begin with the innovation process and not with invention" (p. 68). The divergence and convergence of competence and project relationships, and the focus on projects in which external exploiters and not only internal inventors are involved (i.e., where innovation is needed), are directly related to this view in the relationship-based competence management model.

A good example of the productivity-boosting effects of innovations rather than inventions is the model-based fault diagnosis approach. It was taken in use after expert systems had failed as a technology for making use of knowledge-based techniques in industrial fault diagnosis. The present project portfolio of VTT includes the idea of linking inventions to innovations, but in practice there are still considerable gaps in between.

One of the key aspects of the relationship-based competence management model is the dynamic, simultaneous change of relationships and competence. Law and Callon in Bijker and Law [1992] discuss "the life and death of an aircraft", conducting a network analysis of a project in which the aim was to build a military aircraft (pp. 21–52). The authors use the concepts of *global network*, *local network* and *obligatory point of passage* in between to investigate the project. In terms of the relationship-based competence management model, the global network can be likened to the project portfolio of the R&D supplier, local networks to individual projects, and points of passage to the interplay between the projects.

Law and Callon (page 49) use the dimensions "degree of mobilisation of local network actors" (high-low) and "degree of attachment of actors in global network" (high-low) to characterise the two interrelated networks. If both dimensions are high, the project is "solid, indispensable". If they are low, the project is "weak, disaggregating". In terms of the relationship-based competence management model, the strategic alliance and key supplier situations would contribute to the former both for the supplier and customer sides, whereas the R&D market and resource adjustment situations may easily result in the latter. Although the degree of mobilisation of local network actors would be high – as it was in all projects conducted by the focal code generation researchers – the lack of attachment of actors in the global network will result in divergence. In the code generation case, this happened regarding both customers and the VTT managers, the other key group of internal actors. The internal customers did not adopt code generation as any essential element in their local networks either.

In this study, when building the relationship-based competence management model, some of the key concepts of van de Ven et al. [1999] have been used, such as the converging and diverging behaviour related to innovations. The textbook is based on the famous Minnesota Innovation Research Program carried out during the past seventeen years to answer the question (p. ix) "how and why innovations develop over time from concept to implementation?". This question is quite close to the research problem in this research, and its latter part is directly related to the managerial perspective addressed in this fourth part of the dissertation.

Van de Ven et al. also address the task of "managing relationships during the innovation journey" (pp. 125–148), as none of the innovations that they have investigated was a "self-sustaining, autonomous entity". They argue that it is important to understand the dynamics of relationships, especially how "dyadic relationships of a focal innovative unit become interdependent and influence each other over time". They call the results of such an evolution inter-organisational webs, defined as sets of interconnected, dyadic relationships.

Such a web has been considered in the project portfolio of the focal R&D supplier, and it can also be claimed that the present approach to these relationships at VTT suffers from a fallacy of linear and independent evolution of individual dyads, in which competence should evolve and converge. Van de Ven et al. point out that during the early development stages, interaction mainly takes place among the participants of individual dyads, which is obvious in the R&D market situation. It is critical that relationships would be "established during relatively short divergent periods followed by longer convergent periods to carry out the agreements made" (p. 127). After the fall of expert system technologies, convergence in the fault diagnosis relationships started to emerge and lasted until the first GSM network diagnosis project. In the code generation case, the divergence of the customer relationships – and the competence – never ceased to exist.

The authors use the concepts co-operation, conflict, regulation and competition to classify the behaviour of relationship parties. In this study, similar concepts have been used in the competence evolution framework, but with a simpler view based on mutuality. Moreover, the processes have not been addressed through which co-operation evolves in relationships (cf. Figure 5.1, p. 130 in [Van de Ven et al. 1999]). Instead, a more straightforward view was adopted to describe how the types of parties and the form of interaction affect the evolution. However, the purpose is the same, "understanding and explaining how any one relationship unfolded over time", which "requires looking beyond that individual dyadic IR [inter-organisational relationship] to the larger webs of IRs in which the parties are involved" (p. 146). The point where the web becomes more important than any individual dyad is called the "self-organising criticality" by the authors. Obviously, the identification of such a point would mean a start for the core platform management process.

The concept of path dependence, which is discussed in Håkansson and Waluszewski [1999] and which has already been referred to, may not only restrict but also facilitate technical development. The case discussed by the authors shows how existing resources that have been used in some relationships can suddenly provide for entirely new opportunities, if combined with other resources in new relationships. In other words, some resources the use of which has already been converged, may face new diverging paths. This is, however, "not a smooth process".

Especially at the cross-roads of new paths the "heaviness" of existing resources and relationships may cause problems. The first GSM network diagnosis project offers a good example. The concurrent R&D processes are a means of both managing existing path dependencies (for example, among resources included in the core platform) and introducing new cross-roads (for example, in R&D services delivered to new customers). If the platform reuse and service reusability processes are taken in use, the existing paths must constantly be integrated with new paths. It is important to recognise that even if some competence elements could be reused as such in a new project, in practice they have to be integrated with some other competence elements on the customer side. The R&D supplier thus faces the cross-roads of reusing and renewing its competence continuously, not alone but as part of its customers' competence reuse and renewal processes.

10 CONCLUSIONS

The process and results of this research have been discussed along the way, as well as reflected to what has been done by other researchers. As already pointed out above many times, competence evidently evolves within the context of external project relationships, and not only for such relationships – which was the main assumption included indirectly in my research problem on how competence evolves in project relationships. To solve the problem, it was necessary to find answers to several questions,

1. what does competence mean in the context of contractual project-based research and development (and how to conceptualise both competence and project relationships),

2. how does competence change along time in the focal organisation (and how to make changes explicit),

3. what affects competence evolution (and how to illustrate it in the context of project relationships), and

4. how to manage competence evolution, in practice?

The first question is addressed in the second part of this dissertation, which describes the initial competence evolution framework and demonstrates its use in the analysis of the fault diagnosis case. It lays the foundation for the main conceptual contributions of this research, which cannot be overlooked, although the development of the framework has been based largely on existing work. What has been missing so far is a combination, extension and application of the concepts in the context of contractual R&D.

The second question was certainly the most difficult one to answer. The initial concepts used for making competence and relationships explicit needed to be revised, resulting in a rather complex model of competence evolution. Once again, this model is a combination of existing concepts, which, however, seem to fit together rather well. Using this model, the developments of the code generation case during the past years could be analysed. In all probability, this research is the very first longitudinal and detailed qualitative analysis of competence evolution at VTT Computer technology laboratory and VTT Electronics, the focal organisation.

When addressing *the third question*, it became evident that non-aligned goals and conflicting logic of action made it extremely difficult for an organisation to create core competence, although the original inventors of some competence elements would keep the elements alive through difficult times. Resource ties and activity links can, on the other hand, provide for building core platforms around the portfolio-specific knowledge of specific applications and functional problem solving skills. These are the dimensions emphasised by Elfring and Baven [1996]. The process is affected by changes of generic techniques and technologies, as well as by the success of acquiring new application knowledge, especially if competing solutions have already been developed by the customer's own experts.

Although the framework created in connection with this research was helpful in gaining an understanding of how competence had evolved in the fault diagnosis and code generation cases, this closing part of the dissertation indicated that *the fourth question* could be answered only partially. One of the basic lessons learned along the research is that it is useful to apply analytical means to look back, before rushing into "improving" existing R&D practices. The analytical work took a lot of time and was not by any means straightforward to carry out. Therefore, the fourth part of the dissertation provides only for some initial insight into relationship-based competence management. Further research would be needed to analyse the managerial concepts proposed in this paper. Research and practice should proceed largely in parallel, due to the speed of ongoing developments in contractual engineering research and development.

One of the most important empirical findings of this research concerns the following: Both of the cases studied indicate that the outer context may essentially affect the project purchasing horizons of the customer and thereby also influencing on the realistic project marketing horizon and the evolution of competence. In the fault diagnosis case, the initial project marketing horizon needed to be completely revised due to the fall of expert system technologies. In the code generation case, it was basically the recession and differing logic of action of the two key groups of actors that undermined the possibilities of establishing any red projects, and the blue projects used as a substitute complemented the negative developments from the viewpoint of making use of the competence gained.

It should also be pointed out once again that the skills in specific engineering techniques turned into core rigidity of a kind in both cases. In the fault diagnosis case, the special knowledge engineering techniques on which problem solving was initially based were fortunately not only replaced by newer techniques, but also organised into a pool of alternatives to be flexibly chosen according to customer preferences. In the code generation case, the structured SA/SD system design technique constituted the heart of the whole approach. And it was not replaced until in the late nineties, when it was, apparently, too late.

The external relationships of VTT, although complex, appeared not as overly problematic at the level of individuals. The persons interviewed for the purposes of this study were quite understandable regarding their past disagreements encountered in various projects, if there had been any. However, VTT seems not to be availing itself too much of the changing evaluation of the results of its past relationships, although the contractual agreements of the organisation explicitly include a possibility of interviewing customers several years after the projects have been ended. This opportunity should be utilised to a higher degree, as a customer relationship management routine. Interesting findings would certainly be revealed, and the possibilities of assessing the longer-term effects of competence would be improved. There are few other systematic means available for finding out how the competence at VTT has been delivered to customers in the past.

It is obvious that much more can be done to gain a better understanding of professional service firms. Only a few people have been concerned with such firms in the business research community. It would thus be of great importance to simply make more people interested in the problems faced by professionals in service firms on a daily basis. However, rather than trying to put together any exhaustive list of unresolved research topics, I would like to point out only one key aspect. As indicated above, although the investigations of industrial networks have utilised interpretative and qualitative research methods, researchers have been inclined to take rather objective and rational viewpoints to the network actors. The logic of action has been used in this research to provide for a more subjective perspective to people's behaviour in complex networks. However, it would be interesting to carry out network research from an even more subjectivist perspective by, for example, living as a researcher through certain relationships. This kind of action research should be encouraged, as it is likely to offer a better readiness not only for reacting to, but also for influencing the changing inner and outer contexts of industrial networks. Such an approach would, naturally, strengthen the managerial consequences of network research, towards a constructive point of view.

The author of this study has been involved in the cases studied, but well before they were taken under study. This research has revealed only some of the strongest feelings, motivations and agendas of the interviewees. The task of putting together an entire puzzle from the viewpoints of individuals has been a challenging one. However, even in organisations that create and utilise expert skills, the professional attitude towards work makes it difficult to find out why and how people really affect the developments. To mention only one example, the author had the opportunity of learning quite a lot, not only from all those interviewed during this research, but also about himself. It would be interesting to investigate the driving forces of the individuals involved in the processes. Without such forces, the code generation skills, for example, would have been lost only a few years before their breakthrough may finally take place.

It seems that the need to tackle both the comprehensive economic developments of the outer context and to understand the individuals involved in the inner context may have made quite a number of doctoral researchers developing or applying the network model – including certainly myself – prisoners of contexts. We have been circulating around contexts, too scared to jump right in. A thorough investigation of the outer context (cf. [Lovio 1993]) is likely to require a lot of work and collaboration, to allow the addressing of complex industry-level, nation-wide or even global issues. Similarly, to understand what really goes on in the inner context might be possible only by stepping into the shoes of the persons involved. The front door of the inner context has now been opened, but what has been done is but walking around the entrance hall, where both my worn-out manager's slippers and some inventors' handsome boots were laying. It can be suspected, however, that many other pieces of footwear have been hidden in the closets of the house. Some of them could have taken researchers much deeper into the wilderness of professional R&D services starting right from the backyard.

REFERENCES

Abell, D.A. 1980. Defining the business. Prentice-Hall, Englewood Cliffs, New Jersey, 1980. 257 p.

Alajoutsijärvi, K. 1996. Rautainen pari. Kymmenen ja Valmetin suhde, lähiverkosto ja makrovoimat 1948–90. Jyväskylä Studies in Computer science, Economics and Statistics 31. University of Jyväskylä, 1996. 279 p. (in Finnish)

Alajoutsijärvi, K., Eriksson, P. 1998. Paper life. Making sense of a forest sector triad in its contexts. In: Tikkanen, H. (ed.). Marketing and International Business. Essays in Honour of Karin Holstius on her 65th Birthday. Publications of the Turku School of Economics and Business Administration.

Alajoutsijärvi, K., Tikkanen, H. 1999a. Competence-based business processes within industrial networks. A Theoretical and Empirical Analysis. In: Woodside, A. (ed.). Advances in Business Marketing.

Alajoutsijärvi, K., Tikkanen, H. 1999b. Competence development within industrial networks. Analyzing a case. To appear in the Journal of International Marketing and Marketing Research. 18 p.

Alasuutari, P. 1994. Laadullinen tutkimus. Vastapaino, Tampere. 281 p. (in Finnish)

Anderson, H. 1994. Ett industriföretags omvandling. En studie av Hägglund's förändringsprocess 1922–81 med bas i företags produkter, relationer och resurser. Doctoral dissertation, Linköping University, Department of Management and Economics. 229 p. (in Swedish)

Anderson, E., Coughlan, A. 1987. International market entry and expansion via independent or integrated channels of distribution. Journal of Marketing, Vol. 51, January, pp. 71–82.

Anderson, J.C., Narus, J.A. 1984. A model of the distributor's perspective of distributor-manufacturer working relationships. Journal of Marketing, Vol. 48, pp. 62–74.

Anderson, J.C., Narus, J.A. 1990. A model of distributor firm and manufacturer firm working partnerships. Journal of Marketing, Vol. 54, January, pp. 42–58.

Anon. 1987a. Uuden teknologian tuotekehitys- ja tutkimuskeskus, perustamisselvitys. Teknologian diffuusio, TEDI-87. Projektiraportti n:o 1. City of Oulu. (in Finnish)

Anon. 1987b. Uusia keinoja teknologiayhteistyön tehostamiseksi Oulun seudulla. Teknologian diffuusio, TEDI-87. Projektiraportti n:o 2. City of Oulu. (in Finnish)

Anon. 1987c. Pohjois-Pohjanmaan elinkeinoelämän tietotekniset toiminta-edellytykset. Pohjois-Pohjanmaan Seutukaavaliitto. Publications Series A:84. 61 p. (in Finnish)

Anon. 1989. Oulun läänin teknologiapoliittinen ohjelma. Oulun lääninhallitus. 54 p. (in Finnish)

Anon. 1990a. Valtion tiede- ja teknologianeuvosto: katsaus 1990. Tiede- ja teknologiapolitiikan suuntaviivat 1990-luvulla. 76 p. (in Finnish)

Anon. 1990b. Teknologiaohjelmatoiminnan linjat 1990-luvulle. Komiteamietintö 1990:2. 112 p. + app. (in Finnish)

Anon. 1992a. Teollisuuspoliittinen linjaus. Seteli. 16 p. (in Finnish)

Anon. 1992b. Sähkö- ja elekteroniikkateollisuus. Seteli. 7 p. (in Finnish)

Anon. 1993a. Teknologiakatsaus 1993. Tekes. 77 p. (in Finnish)

Anon. 1993b. Kansallinen teollisuusstrategia. Kauppa- ja teollisuusministeriön julkaisuja 1/1993. Ministry of Trade and Industry. 124 p. + app. (in Finnish)

Anon. 1994a. Suomi tietoyhteiskunnaksi – kansalliset linjaukset. Ministry of Financing. 73 p. (in Finnish)

Anon. 1994b. Osaamisstrategia. Sähkö- ja elektroniikkateollisuuden menestystekijät, avainteknologiat ja osaamisen kehitystarpeet. Seteli. 29 p. (in Finnish)

Anon. 1996. Teknologia 2000. Osaamisella tulevaisuuteen. Tekes. 120 p. (in Finnish)

Anon. 1998. Huippuosaajat maailmalla. Sähkö- ja elektroniikkateollisuusliitto. 21 p. (in Finnish)

Axelsson, B. 1995. The development of network research – A question of mobilization and perseverance. In: Möller, K., Wilson, D.T. (eds.). Business marketing: An interaction and network perspective. Kluwer Academic Publisher, Boston, MA. Pp. 111–137.

Bacharach, S., Bamberger, P., Sonnestuhl, W. 1996. The organizational transformation process: The micropolis of dissonance reduction and the alignment of logic of action. Administrative Science Quarterly, Vol. 41, pp. 477–506.

Bogner, W.C., Thomas, H. 1996. From skills to competences: The "play-out" of resource bundles across firms. In: Sanchez, R., Heene, A., Thomas, H. (eds.). Dynamics of competence-based competition: Theory and practice in the new strategic management. Elsevier Science, Oxford. Pp. 101–117.

Boisot, M. 1994. Information and organization: The manager as anthropologist. Harper and Collins, London.

Boisot, M., Griffiths, D., Moles, V. 1995. The dilemma of competence: Differentiation versus integration in the pursuit of learning. Proc. of the 3rd International Workshop on Competence-Based Competition, Ghent, Belgium, November 16–18.

Bryman, A., Burgess, R.G. (eds.) 1994. Analyzing qualitative data. Routledge, London.

Campbell, N.C.G. 1985. A relationship approach to organizational buying behavior. Journal of Business Research, Vol. 13, pp. 35–48.

Chatterji, D. 1995. Achieving leadership in environmental R&D. Research & Technology Management, Vol. 38, No. 2, pp. 37–42.

Christensen, J.F. 1996. Analysing the technology base of the firm. A multi-dimensional resource and competence perspective. In: Foss, N.J., Knudsen, C. (eds.). Towards a competence theory of the firm. Routledge, London. Pp. 111–132.

Cooper, R.G., Kleinschmidt, E.J. 1996. Winning businesses in product development: the critical success factors. Research & Technology Management, Vol. 39, No. 4, pp. 18–29.

Dubois, A. 1994. Organising industrial activities – An analytical framework. Chalmers University of Technology, Department of Industrial Marketing, Gothenburg, Sweden. 166 p. + app. 1 p.

Dwyer, F.R., Schurr, P.H., Oh, S. 1987. Developing buyer-seller relationships. Journal of Marketing, Vol. 51, No. 2, pp. 11–27.

Easton, G. 1992. Industrial networks: a review. In: Axelsson, B., Easton, G. (eds.). Industrial networks – A new view of reality. Routledge, London, 1992. Pp. 1–27.

Easton, G. 1995. Methodology and industrial networks. In: Möller, K., Wilson, D.T. (eds.). Business marketing: An interaction and network perspective. Kluwer Academic Publisher, Boston, MA. Pp. 411–492.

Easton, G., Araujo, L. 1996. Characterizing organizational competences: An industrial networks application. In: Sanchez, R., Heene, A., Thomas, H. (eds.). Dynamics of competence-based competition: Theory and practice in the new strategic management. Elsevier Science, Oxford. Pp. 184–207.

Elfring, T., Baven, G. 1996. Spinning-off capabilities: Competence development in knowledge-intensive firms. In: Sanchez, R., Heene, A., Thomas, H. (eds.). Dynamics of competence-based competition: Theory and practice in the new strategic management. Elsevier Science, Oxford. Pp. 209–225.

Eriksson, P., Ropo, A. 1995. Managing international brands: A process of competence building. University of Tampere, Series A3: Working Papers 24.

Eriksson, P., Keso, H. 1997. The process of organizational knowledge base change. A longitudinal case study of an aircraft company. Proc. of the Competing through Knowledge Conference, 6–7 May 1997, Bristol.

Eriksson, P., Räsänen, K. 1998. The bitter and the sweet: Evolving constellations of product mix management in a confectionary company. European Journal of Marketing, Vol. 32, No. 3/4, pp. 279–304.

Ford, D., Gadde, L.-E., Håkansson, H., Lundgren, A., Snehota, I., Turnbull, P., Wilson D. 1998. Managing business relationships. John Wiley & Sons, Chichester, UK. 292 p.

Ford, D., Håkansson, H., Johansson, J. 1986. How do companies interact? Industrial Marketing and Purchasing, Vol. 1, No. 1, pp. 26–41.

Foss, N. (ed.) 1997. Resources, firms and strategies. Oxford University Press, Oxford.

Foss, N., Knudsen, C. 1996. Towards a competence theory of the firm. Routledge, London.

Gallon, M.R., Stillman, H.M., Coates, D. 1995. Putting core competency thinking in practice. Research & Technology Management, Vol. 38, No. 3, pp. 20–28.

Gemunden, H.G., Ritter, T. 1996. Managing technological networks: The concept of network competence. In: Gemunden H.G., Ritter, T., Walter, A. (eds.). Proc. of the 12th International Conference on Industrial Marketing and Purchasing, Karlsruhe, Germany.

Ghauri, P., Gronhaug, K., Kristianslund, I. 1995. Research methods in business studies. A practical guide. Prentice Hall, New York. 162 p.

Gressetvold, E. 1998. The network as source for resource development in Nordic VLSI. Proc. of the CBIM/ISBM Conference, Atlanta, GA. 18 p.

Halinen, A. 1994. Exchange relationships in professional services: A study of relationship development in the advertising sector. Turku School of Economics and Business Administration, A-6. 393 p.

Halinen, A., Törnroos, J.-Å. 1995. The meaning of time in the study of industrial buyer-seller relationships. In: Möller, K., Wilson, D.T. (eds.). Business marketing: A relationship and network perspective. Kluwer Academic Publisher, Boston, MA. Pp. 494–529.

Hamel, G., Prahalad, C.K. 1994. Competing for the future. Harvard Business School Press, London. 327 p.

Heide, J.B., George, J. 1988. The role of dependence balancing in safeguarding transaction-specific assets in conventional channels. Journal of Marketing, Vol. 52, January, pp. 20–35.

Hienonen, R. 1997. Elektroniikka- ja sähköalan kehitysnäkymät 1997...2002. VTT Automation. 229 p. (in Finnish)

Hirsijärvi, S., Hurme, H. 1995. Teemahaastattelu, 7th edition. Yliopisto-paino, Helsinki. 144 p. (in Finnish)

Holmlund, M. 1997. Perceived quality in business relationships. Publications of the Swedish School of Economics and Business Administration No. 66, Helsinki. 336 p.

Holmlund, M., Törnroos, J.-Å. 1997. What are relationships in business networks? Management Decision, Vol. 35, No. 4, pp. 304–309.

Husso, K. 1993. Teknologian diffuusio: Tietotekniikan kehittämisohjelman tekoälyprojekti diffuusion näkökulmasta. Technology Development Centre of Finland, Helsinki. 109 p. (in Finnish)

Håkansson, H. 1987. Industrial technological development. A network approach. Croom Helm, London.

Håkansson, H., Snehota, I. 1995. Developing relationships in business networks. Routledge, London. 418 p.

Johanson, J., Mattson, L.-G. 1997. Network positions and strategic action – An analytical framework. In: Ford, D. (ed.). Understanding business markets: interaction, relationships and networks. Second edition. The Dryden Press, London. Pp. 176–193.

Karjalainen, J. (ed.) 1991. Finsoft. Ohjelmistoteknologian ohjelma. Sulautetut järjetelmät 1988–1991. Tekes, Helsinki. 167 p. (in Finnish)

Kirjavainen, P. 1997. Strateginen oppiminen tietointensiivisessä organisaa-tiossa. Teoriaa luova case-tutkimus oppimisesta kahden tietoyrityksen strategisessa kehityksessä. Publications of the Turku School of Economics and Business Administration. Series A-2:1997. 369 p.

Kivisaari, S., Lovio, R. 1993. Suomen elektroniikkateollisuuden merkit-tävien innovatiivisten liiketoimintojen menestyminen 1986–1992. VTT Technology Research Group, Espoo. 51 p. (in Finnish)

Klein, J., Gee, D., Jones, H. 1998. Analyzing clusters of skills in R&D – Core competencies, metaphors, visualization, and the role of IT. R&D Management, Vol. 28, No. 1, pp. 37–42.

Klus, J.P., Markkula, M., Venho, J., Järvenpää, A., Sirkeinen, U., Ahlroos, R. 1985. Effective technology transfer. 128 p.

Knudsen, C. 1996. The competence perspective. A historical review. In: Foss, N.J., Knudsen, C. (eds.). Towards a competence theory of the firm. Routledge, London. Pp. 13–37.

Kurki, M. 1995. Model-based fault diagnosis for mechatronic systems. Technical Research Centre of Finland, Espoo. VTT Publications 223. 116 p.

Laamanen, T. 1997. The acquisition of technological competencies through the acquisition of new, technology-based companies. Helsinki University of Technology, Department of Industrial Management, Institute of Strategy and International Business, Espoo. 285 p.

Leonard-Barton, D. 1992. Core capabilities and core rigidities: A paradox in managing new product development. Strategic Management Journal, Vol. 13, pp. 111–125.

Leppälä, K. 1995. Inside a contract research laboratory. A study of concepts, methods and performance. Technical Research Centre of Finland, Espoo. VTT Publications 227. 250 p. + app. 22 p.

Lewis, M.A., Gregory, M.J. 1996. Developing and applying a process approach to competence analysis. In: Sanchez, R., Heene, A., Thomas, H. (eds.). Dynamics of competence-based competition: Theory and practice in the new strategic management. Elsevier Science, Oxford. Pp. 141–163.

Lindman, M. 1997. Managing industrial new product development in the long run. A resource-based view on new product development and performance. Acta Wasaensia No. 59. University of Vaasa. 336 p.

Lovio, R. 1993. Evolution of firm communities in new industries. Acta Academiae Oeconomicae Helsingiensis. Series A:92. Helsinki School of Economics and Business Administration. 304 p.

Lowendahl, B. 1997. Strategic Management of professional service firms. Handelshojskolens Forlag, Copenhagen.

Lundgren, A. 1992. Coordination and mobilization processes in industrial networks. In: Axelsson, B., Easton, G. (eds.). Industrial networks: A new view of reality. Routledge, London.

Matikainen, E. 1998. Efficient governance of interorganizational business relationships. Helsinki School of Economics and Business Administration. Acta Universitatis Oeconomicae Helsingiensis A-141. 195 p.

Meyer, M.H., Lehnerd, A.P. 1997. The power of product platforms. Building value and cost leadership. The Free Press, New York. 267 p.

Michelsen, K.-E. 1993. Valtio, teknologia, tutkimus. VTT ja kansallisen tutkimusjärjestelmän kehitys. Technical Research Centre of Finland, Espoo. 429 p. (in Finnish)

Miettinen, R. 1993. Methodological issues of studying innovation-related networks. VTT Group for Technology Studies, Espoo. Working Papers No. 4. 61 p.

Miettinen, R. 1996. From research to innovation. The case of biotechnical pulp bleaching. Technical Research Centre of Finland, Espoo. VTT Publications 270. 57 p.

Miettinen, R. 1998. Object construction and networks in research work: the case of research on cellulose degrading enzymes. Social Studies of Science, Vol. 28.

Miller, R. 1995. Applying quality practices to R&D. Research & Technology Management, Vol. 38, No. 2, pp. 47–54.

Mårtenson, G., Otala, M., Wiio, O.A. 1985. Tietotekniikka 1990-luvulla. Sitra. Series B, No. 78. 112 p. (in Finnish)

Möller, K. 1994. Interorganizational marketing exchange: metatheoretical analysis of current research approaches. In: Laurent, G., Lilien, G.L., Pras, B. (eds.). Research transitions in marketing. Kluwer Academic Publishers, Boston, MA. Pp. 347–372.

Möller, K., Wilson, D.T. 1988. Relationship perspective in business marketing: an exploratory contingency framework. Helsinki School of Economics and Business Administration, F-233.

Möller, K., Wilson, D.T. (eds.) 1995a. Business marketing: a relationship and network perspective. Kluwer Academic Publishers, Boston, MA. 635 p.

Möller, K., Wilson, D.T. 1995b. Business relationships – An interaction perspective. In: Möller, K., Wilson, D.T. (eds.). Business marketing: An interaction and network perspective. Kluwer Academic Publisher, Boston, MA. Pp. 23–52.

Möller, K., Wilson, D.T. 1995c. Dynamics of relationship development. In: Möller, K., Wilson, D.T. (eds.). Business marketing: An interaction and network perspective. Kluwer Academic Publisher, Boston, MA. Pp. 53–69.

Nonaka, I., 1994. A dynamic theory of organizational knowledge creation. Organization Science, Vol. 5, No. 1, pp. 14–37.

Nonaka, I., Takeuchi, H., 1995. The knowledge-creating company: How Japanese companies create the dynamics of innovation. Oxford University Press, New York. 284 p.

Pettigrew, A. 1985. The awakening giant: Continuity and change in ICI. Basil Blackwell, Oxford.

Pettigrew, A.M. 1987. Introduction: Researching strategic change. In: Pettigrew, A.M. (ed.). The management of strategic change. Basil Blackwell, Oxford, 1987. Pp. 1–13.

Pettigrew, A. 1989. Context and action in the transformation of the firm. Journal of Management Studies, Vol. 6, No. 24, pp. 649–670.

Pettigrew, A. 1992. The character and significance of strategy process research. Strategic Management Journal, Vol. 13, Winter Special Issue, 1992, pp. 5–16.

Pettigrew, A. 1997. What is processual analysis? In: Ropo, A., Eriksson, P., Hunt, J.G. (eds.). Special issue on reflections on conducting processual research on management and organizations. Scandinavian Journal of Management, Vol. 13, No. 4, pp. 337–347.

Pettigrew, A., Ferlie, E., McKee, L. 1992. Shaping strategic change. Sage, London.

Pettigrew, A. Whipp, R., Rosenfeld, R. 1989. Competitiveness and the management of strategic change processes: a research agenda. In: Francis, A., Tharakan, M. (eds.). The Competitiveness of European Industry: Country Policies and Company Strategies. Routledge, London. Pp. 110–36.

Pfeffer, J., Salancik, G.R. 1978. The external control of organizations: A resource dependence perspective. Harper and Row, New York.

Prahalad, C.K., Hamel, G. 1990. The core competence of a corporation. Harvard Business Review, Vol. 68, No. 3, pp. 79–91.

Reger, G., Schmoch, U. (eds.) 1996. Organization of science and technology at the watershed. The academic and industrial perspective. Physica-Verlag, Heidelberg, Germany. 426 p.

Roberts, E.B. 1995. Benchmarking the strategic management of technology. Research & Technology Management, Vol. 38, No. 2, pp. 18–26.

Rogers, A.D.M. 1996. The challenge of the fifth generation R&D. Research & Technology Management, Vol. 39, No. 4, pp. 33–41.

Rosenberg, N. 1976. Perspectives on technology. Cambridge University Press. Cambridge, United Kingdom. 353 p.

Rosenbröijer, C.-J. 1998. Capability development in business networks. A study of distribution in the fine paper sector in the United Kingdom. Doctoral thesis, Publications of the Swedish School of Economics and Business Administration, No. 69, Helsinki. 255 p.

Sanchez, R. 1997. Managing articulated knowledge in competence-based competition. In: Sanchez, R., Heene, A. (eds.). Strategic learning and knowledge management. John Wiley & Sons, Chichester, UK. Pp. 163–187.

Sanchez, R., Thomas, H. 1996. Strategic goals. In: Sanchez, R., Heene, A., Thomas, H. (eds.). Dynamics of competence-based competition. Theory and practice in the new strategic management. Elsevier Science, Oxford, 1996. Pp. 63–83.

Sanchez, R., Heene, A., Thomas, H. (eds.) 1996a. Dynamics of competence-based competition: Theory and practice in the new strategic management. Elsevier Science, Oxford. 403 p.

Sanchez, R., Heene, A., Thomas, H. 1996b. Introduction: towards the theory and practice of competence-based competition. In: Sanchez, R., Heene, A., Thomas, H. (eds.). Dynamics of competence-based competition. Theory and practice in the new strategic management. Elsevier Science, Oxford. Pp. 1–35.

Sanderson, S.W., Uzumeri, M. 1997. The innovation imperative. Strategies for managing product models and families. Irwin Professional Publishing, Chicago. 218 p.

Schumann, P.A. Jr., Ransley, D.L., Prestfood, C.L. 1995. Measuring R&D performance R&D. Research & Technology Management, Vol. 38, No. 3, pp. 45–54.

Seppänen, V. 1995. Asiakaspalveluprosessien strateginen kehittäminen VTT:n tutkimusyksikössä. Executive MBA thesis, University of Oulu, Continuing Education Centre. 79 p. + app. (in Finnish)

Seppänen, V., Isomursu, P., Kähkönen, A.-M., Oivo, M., Perunka, H., Pulli, P. 1996. Strategic needs and future trends of embedded software. Tekes Technology Review 48/96. Technology Development Centre of Finland, Helsinki. 103 p.

Seppänen, V., Alajoutsijärvi, K., Kurki, M. 1998a. Competence-based evolution of contractual R&D relationships. Technical Research Centre of Finland, Espoo. VTT Publications 346. 70 p.

Seppänen, V., Kurki, M., Alajoutsijärvi, K. 1998b. Competence-based evolution of contractual R&D relationships. Proc. of Information Systems: Current Issues & Future Changes, Joint Working Conference of IFIP8.2 and IFIP8.6, Helsinki, December 10–13, 1998. IFIP, Laxenburg, Austria. Pp. 197–214.

Seppänen, V., Alajoutsijärvi, K., Eriksson, P. 1999a. Projects or products: seeking for the business logic of contract R&D. Technical Research Centre of Finland, Espoo. VTT Publications 392. 110 p. + app. 104 p.

Seppänen, V., Alajoutsijärvi, K., Eriksson, P. 1999b. Projects or products – Seeking for the business logic of contract R&D. Proc. of the 15th Annual IMP Conference, Dublin, Ireland, September 2–4, 1999. 10 p.

Silverman, D. 1993. Interpreting qualitative data. Methods for analysing talk, text and interaction. Sage Publications, London.

Sipilä, J. 1992. Asiantuntijapalvelujen markkinointi. Weilin+Göös, Jyväskylä. 386 p. (in Finnish)

Smith, C.B. 1981. A guide to business research. Developing, conducting and writing research projects. Nelson-Hall, Chicago. 190 p.

Sparrow, P. 1992. Building human resource strategies around competencies: A life cycle model. Working paper No. 235. Manchester Business School.

SRI 1996. A world-wide study of technology and innovation management. Survey results. SRI Consulting, Menlo Park, CA. 29 p.

Tallman, S., Atchison, D.L. 1996. Competence-based competition and the evolution of strategic configurations. In: Sanchez, R., Heene, A., Thomas, H. (eds.). Dynamics of competence-based competition: Theory and practice in the new strategic management. Elsevier Science, Oxford. Pp. 349–375.

Teece, D.J., Pisano, G., Shuen, A. 1992. Dynamic capabilities and strategic management. Working paper, University of California at Berkeley.

Thomas, H., Pollock, T., Gorman, P. 1999. Global strategic analyses: frameworks and approaches. Engineering Management Review, Vol. 27, No. 4, pp. 73–84.

Thibaut, J.W., Kelley, H.H. 1959. The social psychology of groups. John Wiley & Sons, New York.

Tiittula, P. 1994. Farewell to bureaucracy: Technical Research Centre of Finland as a pathfinder in management change. Acta Academiae Oeconomicae Helsingiensis. Series A:95. The Helsinki School of Economics and Business Administration. 247 p. + app.

Tikkanen, H. 1997. A network approach to industrial business processes. A theoretical and empirical analysis. Publications of the Turku School of Economics and Business Administration. Series A-7:1997. 227 p.

Tikkanen, H. 1998. Research on international project marketing. A review and implications. In: Tikkanen, H. (ed.). Marketing and International Business – Essays in Honour of Professor Karin Holstius on her 65th Birthday. Publications of the Turku School of Economics and Business Administration. Series A-2:1998. Pp. 262–285.

Tikkanen, H., Alajoutsijärvi, K. 1998. Customer satisfaction in industrial markets: contextual understanding as a key for effective management. Proc. of the Annual CBIM/ISBM Meeting, Penn State University, Atlanta, Georgia, January 17–20. 19 p.

Tipping, J.W., Saffron, E., Fusfeld, A.R. 1995. Assessing the value of your technology. Research & Technology Management, Vol. 38, No. 5, pp. 22–39.

Uusitalo, O. 1995. A revolutionary dominant design. The float glass innovation in the flat glass industry. Acta Universitatis Eoconomicae Helsingiensis, Report A-108. Helsinki School of Economics and Business Administration. 215 p.

Ward, T., Mellor, S.J. 1985–1986. Structured development for real-time systems. Yourdon Press, New York. Vols. 1, 2 and 3.

Wernerfelt, B. 1984. A Resource-based view of the firm. Strategic Management Journal, Vol. 5, No. 5, pp. 171–180.

Werner, B.M., Shouder, W.E. 1997. Measuring R&D performance – State of the art. Research & Technology Management, Vol. 40, No. 2, pp. 34–42.

Williamson, O.E. 1985. The economic institutions of capitalism. The Free Press, New York.

Wilson, D.T., Mummalaneni, V. 1986. Bonding and commitment in buyer-seller relationships: A conceptualization. Industrial Marketing and Purchasing, Vol. 1, No. 3, pp. 44–58.

Yin, R.K. 1988. Case study research. Design and methods. Revised edition. Applied Social Research Methods Series, Vol. 5. Sage Publications, Newbury Park, CA. 165 p.

APPENDIX 1. THE FAULT DIAGNOSIS STORY

In the late eighties, a small knowledge engineering research group of about five persons was established at the focal organization by separating it from a larger software engineering research group. One of the main reasons for this reorganisation was the ongoing interest in “intelligent” knowledge-based systems. Interest was growing rapidly world-wide. The group was established mostly to support the launching of new research and development projects, financed by Tekes, a newly established public funding body, and by individual industrial companies. Another important goal was to help a number of researchers in gaining expertise in this new area. None of the original members of the group had much formal training or previous experience in knowledge engineering.

Knowledge engineering research

Since knowledge engineering techniques were being actively researched everywhere, industry also became interested in adopting them. This interest was indicated by the fact that bigger companies especially were hiring knowledge engineering experts, or were training their personnel in these techniques. Industrial professionals who needed to know more about the techniques were actively seeking contacts with researchers. The initial relationships of the focal research group started to emerge in the late eighties, involving quite a number of industrial professionals who wished to associate themselves with researchers. In connection with a research program launched by Tekes in the mid-eighties, the focal organization invested in expensive system development tools, together with some other VTT institutes, universities, and a few industrial organizations.

Use of the special tools contributed in bringing VTT knowledge engineering experts and their customers closer together, thus strengthening their relationship. One of the goals was to develop knowledge-based applications by using these tools, and then port the solutions to traditional computer systems. This “embedding” of knowledge-based systems was seen as becoming a special capability for the focal organization, as it had dealt with so-called embedded computer systems for many years.

The approach was also accepted by industry, as it was believed that knowledge-based systems needed development using specific techniques and tools, after which the developed solutions could be transferred into more conventional computers for use. However by the mid-nineties, most of the special tools had completely vanished, as they had been replaced by general-purpose computers. The embedding problem had also disappeared. Investments in such special tools had become obsolete.

A number of process and machine automation firms had taken part in a Tekes research program, and had thus become interested in knowledge-based techniques. Yet only a few automation system professionals saw these techniques as a means of making products more “intelligent”. Instead, they faced such problems as how to manage computer-processed data incorporated into modern, automated machines. By taking into account the practical requirements of machine automation applications, VTT researchers proposed knowledge-engineering techniques as a means of solving data management problems. Most machine manufacturers had already become accustomed to computer control in their products. However, features like intelligent data management and fault diagnosis were still seen as being rather remote problems by quite a number of manufacturers, who wished to focus simply on computerising the basic control functions of their products. Thus the interests of Finnish machine manufacturers in knowledge-based fault diagnosis heavily depended on the technological maturity of their products.

At the turn of the decade, the focal organization eventually managed to establish projects with a few machine and process automation companies, and started to develop knowledge-based fault diagnosis features for their products. These companies had already developed quite sophisticated computer-based control systems, several commercial technologies for implementing such systems were available, and – most importantly – the end users of automated machines were increasingly interested in fault diagnosis. Another reason for the focal organisation’s success in establishing fault diagnosis projects without much previous research might be due to certain industrial partners of joint fault diagnosis research projects carried out in the late eighties, who were not satisfied with the results.

The focal research group was not very experienced in process and machine automation applications, while some other VTT institutes had specialised more in these fields of engineering. Certain technologies used to implement automation systems were also relatively unknown to the focal personnel. The only possibility then was to market knowledge engineering techniques as a means of solving fault diagnosis problems to machine manufacturers interested in providing such features to their end users. This strategy proved to be a considerable success. Through literature studies, the focal research group had already recognised fault diagnosis as a potential area for use of knowledge engineering techniques.

The VTT employees involved in the early fault diagnosis projects could be characterised as knowledge engineers. The researchers did all the work together with the customers' knowledge engineers, if indeed there were any. Related work documented in literature at that time followed the same approach: individual knowledge engineers or small teams created intelligent systems by utilising so-called “expert knowledge”. The phrase “knowledge acquisition” was used to describe the capture of application-related expert knowledge. Parts of this knowledge were implicit, others were included in automation and process design documents for example.

The knowledge was not so much learnt by the VTT personnel, but rather captured, explicitly represented, and utilised via the expert system development tools. The explication of that knowledge through application expert interviews and other knowledge acquisition methods was encouraged by the generally accepted philosophy of building knowledge-based systems. Application experts, such as process and automation engineers, were usually not familiar with knowledge engineering techniques, and were thus not able to build and maintain knowledge-based systems on their own. This also meant that the VTT knowledge engineering researchers were quite easily able to join forces with their customers' knowledge engineers, if the customer company had such personnel, as well as to exploit techniques in collaboration to learn how to use them in industrial applications.

VTT was aiming to combine knowledge engineering skills with application-specific knowledge. It started to develop its knowledge-based fault diagnosis capabilities especially for process and machine automation applications. If the types of knowledge-based systems developed during the late eighties had really become an industrial practice, a possible evolution of this state of affairs could have led to customers' knowledge engineers competing with focal researchers. However, by the mid-nineties, the early systems had failed to fulfil their promise, due in many cases to the lack of understanding of the real problems in need of a solution. New directions needed to be taken both by VTT and by industry.

Addressing automation companies

One of VTT's additional strengths in fault diagnosis projects was its capability for modelling computerised control systems, experience based on years spent in specifying and designing computer systems for various application types. Combining this expertise with knowledge engineering skills, and with the accumulating experience of customers' fault diagnosis problems clearly brought a fresh perspective to fault diagnosis research.

Instead of building exotic, knowledge-based models for fault diagnosis, conventional computer system modelling techniques could now be used. In other words, instead of using special knowledge-based models, "ordinary" engineering systems models were used for fault diagnosis. Special knowledge-based model management was complicated even for knowledge engineers, not to mention control system designers. Thus the use of conventional computer system models in fault diagnosis greatly helped application experts. With regard to knowledge-based fault diagnosis features, responsibility was moving towards their own core competence in engineering. In practice, this meant that relations between fault diagnosis researchers and customers' application engineers became more important than ever. Accordingly, relations with customers' knowledge engineers became much less important than before. Therefore, relationships would weaken with those customers who had employed knowledge engineers, and with whom the initial fault diagnosis projects had been carried out.

In some cases they vanished altogether, due to the fall of industrial interest in knowledge engineering techniques. The reason for the fall was that the techniques had not resulted in many practical industrial applications after the first three or four years of research. This fall was a global phenomenon. An increasingly important reason for this development was the fact that the justification for the special tools used to develop knowledge-based systems had disappeared. As a result, relations between VTT researchers and industrial knowledge engineers were primarily replaced by those with industrial process and automation engineers.

Lower labour expenses and a narrower focus on knowledge engineering techniques had favoured relationships between universities and industry during the early days of joint knowledge engineering research projects. By the mid-nineties, VTT’s combined computer system modelling, knowledge engineering, and project management skills appeared to be more important for industry, as fault diagnosis systems awaited implemented as part of industrial products, not merely prototypes.

Seen from the focal organisation’s perspective, one of the key results of this period was the laying of foundations for the understanding of the whole fault diagnosis problem solving chain. This understanding was made explicit through tasks required for developing practical fault diagnosis systems. Moreover, as the role of VTT in this problem solving process become clearer, it was easier for both the researchers and the customers' application experts to see what resources and skills each party had brought to the process. Previously, the roles of VTT and industrial knowledge engineers had overlapped and were sometimes competitive, especially when pre-production prototypes had been built instead of commercial products.

Fault diagnosis platform

After several contractual fault diagnosis projects had been carried out for automation companies, it was discovered that not only the tasks of the basic problem solving process but also its results, called diagnosis functions, were rather similar. A researcher, who had managed many fault diagnosis projects and later became head of the focal research group, studied process and function. *Data acquisition, fault detection, fault localisation, fault explanation,* and *fault recovery* functions were identified. The problem solving process was streamlined, and functions gradually evolved towards a fault diagnosis platform. Parts of this platform existed physically as software modules that could be ported to different implementation technologies. At first, the platform had existed merely in the form of marketing overhead slides of the focal research group.

Models of customers’ applications were designed anew in each project, and the system implemented by using technology favoured by particular customers. The functions and the problem solving process could be reused. The platform included implementation technologies, fault diagnosis functions supported by certain knowledge engineering techniques, and application models. The phrase “model-based fault diagnosis” was adopted to describe the approach.

Fault diagnosis systems were also built in the mid-nineties by several other VTT institutes. However, their projects emphasised diagnosis of lower-level functions, such as real-time data acquisition. These functions were often closely integrated with some new implementation technologies and were developed in joint research projects. Apparently, the projects did not adhere to any problem solving process based on given system design tasks. Having already dealt with practical fault diagnosis problems for several years, the focal organization aimed for solutions independent of application and technology framed by a set of system design tasks.

The customers appreciated this strategy of combining their own application expertise with fault diagnosis and computer system modelling skills. By the late nineties, the focal organization had become quite competent in solving practical fault diagnosis problems in the domain of automation applications. It had rather wide-ranging capabilities and an established design process for solving such problems. Its role was clearer than that of some other parties interested in fault diagnosis. In particular, the role of the application experts at other VTT institutes dealing with fault diagnosis, overlapped with the roles of industrial professionals.

New applications

After the fault diagnosis problem solving process and the platform had been used in several contractual projects, it was realised that, apart from automation, there might be many other application areas in which fault diagnosis was needed. Mainly because of its growing business volume, one of the application areas studied was telecommunications. It seemed that the acquisition, monitoring, and analysis of alarm data in telecommunication networks was a fault diagnosis problem, for which system design tasks and platform functions could be used.

However, both this application and the technology used to implement network devices were different from that of automation applications. Although VTT had close contacts with certain telecommunication network equipment manufacturers, there were others with whom it had practically no contact. Moreover, there were no relationships with network equipment end-users, i.e. telecommunication operators. This posed problems in gaining necessary knowledge of the new application domain, especially since telecommunication networks are complex assemblies of interrelated equipment and systems. For example, although an automated machine is a multi-technology system, it is usually a single product sold by a individual company and designed by a fairly small team of engineering and application engineers. There is only a limited number of design documents to study and experts to interview, and both are readily available.

On the contrary, the development of a telecommunication network may involve hundreds of different engineering professionals, as well as many distinct companies, business units, and subcontractors. Several different operators, i.e. customers of the equipment manufacturers, may also be involved in the development. Certain system features that cross individual equipment boundaries may be very poorly documented.

Within this new context, the focal organization decided to address case-based reasoning, a new knowledge engineering technique. A practical reason for this decision was the fact that the first fault diagnosis project involving telecommunication networks was part of a new joint national research program on intelligent systems. As it was economically feasible for VTT, the project was organised as a contractual consortium, so that three different business units of the same company acquired financing to cover part of their expenses. The work was carried out by VTT, since it had originally marketed the topic to these companies studied by a researcher, who had managed many fault diagnosis projects and later became head of the focal research group.

VTT was unable to name any single person, who would be familiar with the application as a whole. Moreover, there were no individual models of the network available, nor even any clear ideas on how the network should be modelled. Operators were not involved in the project, and the equipment manufacturers' knowledge of the end-users' needs could not easily be conveyed to the researchers.

Project emphasis focused largely on selecting the appropriate means for network modelling. The benefits of the fault diagnosis problem solving process and the platform functions could not be shown, nor even the benefits of integrating the case-based reasoning technique and application knowledge for fault diagnosis. The project result was delivered on schedule and according to budget, but it was just a demonstration system with no interfaces to real network equipment. All this, together with some managerial problems in carrying out the project, caused a conflict that almost led to interruption of the work. From VTT’s point of view, this might have resulted in the loss of this application domain for several years.

At this stage, the focal organization was successfully carrying out considerable fault diagnosis projects for both machine and process automation customers. Some of these projects were similar types of large consortia as in the network diagnosis project, in which several different factory sites and research partners were involved, as well as public funding bodies. Some new techniques were also applied in these projects, and the same expensive knowledge engineering tool was utilised. However, a remarkable difference was that many of the employees involved in these projects had already gained experience of automation applications, whereas none of the persons involved in the network diagnosis project had much telecommunication engineering expertise. Moreover, several basic functions of the fault diagnosis platform and the system design tasks could be reused.

APPENDIX 2. THE CODE GENERATION STORIES

In the following, the original case summary written by Veikko Seppänen is accompanied with the *story of the code generation researchers*, created on the basis of interviews and a group rehearsal.

Code generation research and development started at TKO in the mid-eighties. There was a strong, global belief among software engineering researchers that the so-called automatic programming techniques would become practical within a few years. At TKO it was thought that this would be the case also for the development of embedded software. The problem of producing computer software automatically from higher-level system models is known from the very early days of computing. The so-called high-level programming languages had been developed in the sixties and seventies to help programmers to develop software by using higher-level concepts than those directly related to computer hardware. Many kinds of compilers, software tools for generating executable machine code from programming languages, had been available since the sixties.

The input language of compilers, understood and produced by programmers, is called source code. The output language of compilers is executed by computer hardware and called object code. The tools needed in writing source code are, typically, called editors. Different kinds of source code manipulation tools can be packaged into programming environments. Tools needed for managing object code are debuggers, whereas testing tools are used to facilitate executing the code and analysing the execution results. A typical programming task would thus involve the cycle of creating a source code, compiling it into an object code, and debugging the execution results. Many compilers and debuggers are dependent on hardware, the physical computer system that executes the object code. Therefore, software tool vendors sell families of tools tailored to certain computer hardware.

In the seventies and early eighties researchers were trying to “extend” compilation techniques to even higher abstraction levels, to support code generation from even more abstract software modelling concepts. Some researchers and commercial tool vendors were wishing to integrate high-level code generation techniques into more comprehensive Computer-Aided Software Engineering (CASE) tool environments, which emerged in the eighties. Another approach to code generation was to build pre-compilers, tools producing source programs which can then be compiled to object code by means of existing compilers. High-level languages, especially the C programming language, started to become popular in the embedded software development community in the eighties.

The first CASE tools had also been taken into use by embedded software developers. These tools usually supported certain graphical modelling languages and methods used for software development. One of the most popular methods was SA/SD.

A version of the SA/SD method suitable for modelling real-time embedded systems was proposed in the early eighties, soon becoming highly popular. A Finnish tool vendor developed and sold a commercial tool supporting this method. Textbooks and courses on the method became available, sometimes organised around the use of certain tools. A typical software development task would include, firstly drawing graphical models of the behaviour of the system according to the method and perhaps using a CASE tool, and then editing, compiling and debugging the corresponding executable program by means of a C programming environment, for example. A code generator associated with a CASE tool would have automated this task by producing either the source code or the object code from the graphical system models.

At TKO, people had also become interested in SA/SD. In the mid-eighties, in a very short period of time, almost all software engineering researchers took courses on the use of the method, and were applying it in several industrial embedded software development projects. Industrial practitioners took the same course of action, and the Finnish CASE tool vendor succeeded in selling its tool to many of the actual and potential client companies of TKO. Thus, SA/SD based embedded software development emerged rapidly. As researchers, TKO persons became interested in the problem of extending the method and automating the CASE tools that supporting it. One of the ideas for extension was code generation from high-level system models into source code. System modelling notations would be used as a kind of high-level graphical programming language, and the code generator would serve as a pre-compiler for producing embedded software. A pre-study project called Speco on code generation was carried out for an industrial client (firm K) in the mid-eighties, but the input language was not SA/SD and the generated code was not the C programming language. The customer was also not fully satisfied.

This is interesting. I newer fully realised it, but perhaps there was some kind of conceptual fathering away between [the R&D manager of the firm K] and us … First of all, the problem was very complex. [The R&D manager] was wondering why I was proceeding that slowly ... the plan might have been 18 man-months for Speco as a whole. [He] would have needed some kind of journal of what we were doing, and he always had proposals for all problems. I could not write down any kind of journal … At the beginning, I remember that we were having meetings with the walls of the room filled with papers on the semantics of the design elements …We were in the basement of the TKO building then. We were solving the problems at a hectic pace. Minute after minute we were trying to solve some new problem with [yet another researcher]. [The R&D manager] would have needed a detailed journal on how the problem solving process was proceeding … to see, if he was getting some value for his money … Yes, and the preparation of a journal would have required a video recorder, to prevent disturbing the problem solving process … Returning back to my problems with [the R&D manager]. We were both software guys. He could just come to me and say that a particular problem could be solved in that way, and I, after having thought about it for a month, said no, it was not possible … [He] wanted to be right.

A decision was made to propose a code generation research project to the Finsoft software engineering research program established by Tekes, the engineering research funding body, in the late eighties. The researchers closely involved in the Speco project supported this proposal. The proposal was written on the basis of the original ideas of these persons rather than, for example, on the analysis and extension of existing results of automatic programming research. Another possible approach would have been to make a literature survey and to propose how the existing techniques could have been modified for the embedded systems domain.

We performed an extensive literature survey on automatic programming and code generation in the Speco project and collected one big folder full of references. We had become familiar with formal methods for modelling parallel systems in a licentiate course. [The R&D manager of the firm K] presented us the refinement and transformation concepts for Draco. The problem with the traditional approach was its tree structure, it would be better to describe parallel systems as networks.

Now, the idea was more practical in the sense that the proposal addressed the problem of how to transform SA/SD models into C source programs. Both the input and output languages of a tool that would support this transformation were well known to the embedded systems software design community. The proposed code generator would obtain the SA/SD system models from a CASE tool and produce a C source program for a compiler, which in turn would produce the corresponding object code.

I had evaluated related systems, though in a limited way, but using the 'hands-on' style. I surveyed the Draco and Refine transformation tools. I was very interested in Draco, for example, until I realised its problems. On the other hand, I found out its benefits, too. Draco's refinements are actually code components, which the system has been built of. The main obstacle of Draco was addressed by Reino Kurki-Suonio [one of the best known Finnish software engineering professors] at some meeting. With Draco, it is extremely hard to define domain-specific languages. Moreover, the principle of defining the semantics of languages was not good in Draco. The definition of semantics was done by making refinements. In practice, the definition should be based on a more formal model. Refine was based on a so-called wide-spectrum language. This was a strength and a weakness at the same time. The maintenance and management of the language concepts was rather problematic. Yet, I obtained ideas from Refine that were later implemented in ReaGeniX 1.0.

A number of automatic programming researchers had proposed special input languages for code generators so as to generate a full program code. One could say that no practical, industrially usable code generators existed for this reason. Practitioners were not using the kinds of abstract language notations by means of which automatic programming researchers had produced programs. Moreover, many of the languages for which small-scale research prototypes of code generators had been developed, were not suitable for programming embedded systems.

We used Prolog instead, which has actually turned out to be very good for the purpose. Refine, for example, includes a number of features similar to those in Prolog. In Draco, the same features have been implemented by using the Lisp language.

In this regard, the starting point of the code generation research at TKO was quite practical and promising. It was believed that the problem to solve was how to define and implement a mechanism – a code generator tool – that could transform SA/SD models into C source programs. The input and output languages and the technologies supporting the manipulation of these languages existed and could be used in the development of the code generator. It was thus believed, based on the experiences gained from the Speco project, that the project would succeed in producing an industrially usable code generator. Such a tool would have been a remarkable breakthrough indeed.

The persons behind the proposal pointed this out, their aim was to introduce the world's first embedded systems code generator. It could really change the way practitioners were working. This belief was shared by quite a number of industrial embedded systems practitioners. More than ten firms became interested in the proposal, joining the venture as follow-up participants paying a small fee to be able to use the results. A Finnish CASE tool vendor was one of these firms. The Sokrates project was launched in 1988 and it lasted for three years, as part of the Finsoft research program.

Making of the invention

The Sokrates project, although financed by Tekes, by TKO and to some extent also by the participating companies, was carried out by the researchers who initially proposed the project and then became its key resources. The researchers designed the technical specifications of the code generator based on their own original ideas, and established a research environment where the generator would be developed. This environment, a workstation computer, was physically separated from the internal computer network of TKO, so that no one else could disturb the ongoing work. This decision was made by the project group and accepted by the managers of TKO. The development of the code generator did not require any modifications to the CASE tool which was chosen for the task of producing SA/SD-based system models. There was thus no need to establish any particular co-operation relationships with the Finnish or other tool vendors.

[This Finnish CASE tool vendor, which] took part in the steering group [of Sokrates], did not, however, want to be involved in any real co-operation within the project.

However, the SA/SD method needed to be extended, as it was described in the literature. New development tasks and new kinds of models were defined, according to which the input of the code generator would be produced. The main reason for this was that the original method was meant to be read and understood by software engineers, not by a computer.

The original method was, therefore, not rigorous enough for generating program code. In other words, the rather informal and abstract graphical modelling language was redefined and extended to a rigorous programming language. The method and the language were defined through the project and named as Sokrates-SA. A part of Sokrates-SA was a subset of the Ada programming language, popular in American embedded systems military applications, but not used in practice in Finland. The manager of the project gave courses on the redefined method not only to the industrial participants of the project, but to a great number of other embedded systems practitioners and software engineering students as well. The courses were generally considered useful.

The code generator, as a tool, affected both the development process of embedded software and the kinds of software solutions produced in the process. It was not directed to any particular application domain, such as telecommunication or automation, but was rather meant to serve the needs of generic embedded software engineering. However, the input and output languages were fixed, and therefore also the computer hardware, for which source code was be produced.

New input and output languages would have required developing new code generation rules, as well as adding links to new CASE tools and compilers. The rules were, however, not entirely hard coded into the generator. Although the code generator was designed as a generic software engineering tool, the developers did not establish many relationships to other researchers, while using the results produced by these in developing generic automatic programming tools. Most of the existing tools were based on different input languages, or were rather useless in generating full embedded source code. This situation led to the fact that there was no rigorous theory of the generic code generation technique that was implemented, which could have been scientifically evaluated by other researchers dealing with similar problems.

Well, there was no such theory anywhere in the world. [Two code generation researchers] had made an attempt at publishing the principles of the coding rules, but I have to admit that the paper was not too well written (due to the lack of training and experience), which is why it was not published … Our approach was commented by [a colleague] saying that "we raised the bar so high that it was easy to go under it" … [Another researcher claimed that] No, his comment was that "you will find the highest position of a fence to go under it".

Our approach was all the time that if there was some theoretically interesting problem, we would try to figure out how to do things without solving this theoretical problem. Moreover, we all have been developing these things further afterwards. For example, we all knew what objects were, although they were not included in the approach. The reason was that, in our opinion, objects would not bring anything new to solving concurrence problems. We were reading up-to-date object-oriented literature, but its message was that the operating system would solve all the problems.

Attempts were made at publishing some of the code generation principles in the early phase, but the papers were rejected by reviewers.

The code generation technology was published in the STEP-90 conference.

This consolidated the view of the code generation researchers that they were producing something that was unique, a comprehensive approach that was different from all the other approaches. Later, however, a few papers were published on parts of the work. There was a vivid discussion within the Sokrates project on which case application the code generator would be applied to. Various proposals were made, the project, however, decided to build its own test application, a toy elevator constructed from Lego blocks.

Later, during Sokrates, when I went to [visit the firm K] it came to a conflict . They had started a new generation [product] development, new chief designers had come with a new software architecture. [The representative of the firm K in the steering groups of Sokrates] and I were talking to the chief designers trying to offer a Sokrates case, but they reacted very strongly, saying that if the Sokrates case were implemented, they would leave [the firm], or something like that… The case would have involved a small processor, on which they had spent dozens of man-years.

The elevator controlling case had been used as a classical embedded systems text book problem. The project group decided to show how it could be solved by means of the Sokrates-SA method and the code generator. A summer trainee implemented the elevator, which was controlled by an industrial PC computer executing C programs.

Since, despite many requests, we did not manage to get any pilot case from the members of the steering group, the Lego elevator was chosen as a demonstration. We had made an alternative plan [to develop the Lego elevator] for this situation. We activated the alternative plan, because the deadline for starting to work for the case study was coming closer. The Lego elevator was chosen as the alternative for the following reasons: its control was not a trivial matter, including some real problems on concurrence and real-time issues; the problem domain was close to the businesses of the steering group members; I was familiar with the elevator control problem; the elevator would provide good demonstration equipment; the elevator could be developed and tested in a laboratory without any extra travel; damages resulting from errors would be small, and, finally, it would be available as a test environment after the project.

The demonstration succeeded, although rather many of the industrial participants of the project and other researchers were suspicious of using Ada as a part of the Sokrates-SA modelling language.

The reason for choosing Ada was to provide ways of generating different kinds of code using a single input language, and the purpose was also to do research on system analysis – and Ada provided better means of satisfying these needs than C. Yet, I overestimated the increase of the use of Ada. I did the Ada parts and noticed that it was completely senseless to do all the features with Ada. I was never able to get them [the other code generation researchers] to believe that.

A researcher not involved in the code generator project had developed a simpler code generator for one of the key customers of TKO, who took part in the steering group of the Sokrates project. This approach was based on transforming certain parts of the SA/SD models directly into C. It was proposed by the Sokrates steering group and the line managers of TKO that the developers of the code generator would follow a similar path, but the proposal was rejected. This, in practice, prevented the use of the developed generator by the mentioned customer company.

I had taken a look at [the other generator], too. It produced [C] code only from state machines and did not have much for handling data flow diagrams – although the problem in generating code for real-time systems is the management of concurrence [designed by using data flow diagrams in the SA/SD method]. At that time Sokrates was not suitable for letting C code go from specifications to the implementation [as in the other generator].

After demonstrating the generator in the toy elevator example, the researchers talked to the Finnish CASE tool vendor for joining forces to commercialise the generator. The discussions failed to produce any co-operation, the tool vendor had already started its own code generator development and did not wish TKO to become involved.

But the vendor did not admit it. It was clear that [the managing director of the firm] did not want that.

Discussions with other, American vendors, did not result in co-operation either. Yet another line of co-operation was pursued, namely the generation of a hardware description language from SA/SD models instead of C. Also these discussions failed, as the hardware engineering researchers considered the Sokrates-SA modelling language too complex for their needs and were not satisfied with the results of an experiment that had been made. Instead, they started to co-operate with the local university and developed a simpler SA-based notation, upon which hardware description programs could be generated. A university researcher later established a small company to commercialise a tool supporting this approach.

Did the company ever sell the generator? Were there any continuing projects? By the way, VTT advertised the generator in its brochures even before the company was established.

The explicit code generation rules aroused interest in a number of the industrial participants of the Sokrates project. One of them was using so-called programmable logic controllers instead of microprocessors, for implementing distributed automated production lines. In addition, a set of code generation rules was developed for the programming language of the controllers by a student, and the rules were taken in use by the same company. As opposed to the key customer mentioned above, this company was dealing with automation applications instead of telecommunication, the role of software being a supporting technology for the applications.

The emphasis of product development in the company was on other engineering design domains, such as mechanical, process and automation engineering. Most of the company's software designers were lower-level engineering graduates, while subcontracting services were used for implementing automation systems and machine control programs.

Yet another side-track of the code generator development was taken in use by this company, a communication protocol software package developed by the code generation researchers. This software was originally built to demonstrate that a ready-made, generic communication software package for distributed systems could be produced using Sokrates-SA. The protocol software was, most likely, used by the company as a replacement for commercial communication software packages, because it was modelled by using the same Sokrates-SA method as some other parts of the software.

Actually, this is an incorrect statement. Code generation itself is a side-track of a system design philosophy. Another side-track of the philosophy is the communication protocol package. The Sokrates group was trying to emphasise the management of concurrence. The story does dot address this at all. Concurrence was a topic which also interested me and [a colleague]. I've already said that it was not made explicit enough in the story that Sokrates really was a way of solving concurrence problems. This topic was always present in the discussions: how to help designers to manage concurrence. This is not obvious at all in the story, the topic has been treated more as the development of a yet another CASE tool.

Packaging of the results

The Sokrates project ended without any commercialisation or large-scale use of the code generator – rather typically of many research projects during the late eighties. It was more disturbing for the managers of TKO, however, that the only subsequent project based on the results was the development of special code generation rules for the machine automation firm, mentioned above. This project was small, only about two man-months. No further research projects were established, perhaps due to the fact that the original manager of the code generator project had decided to join an industrial company at a late phase of the project, and the new manager did not have enough time to plan any continuing projects.

The generator was a mammoth, it would not have been used by anyone then. It was just a prototype, a demonstration that code generation was possible. Nothing else. It was not even a prototype, just a demonstration. [The firm W] used an 8051 processor, they wanted to be a leading edge firm, they were just looking around. Many others did not have the applications for which the results could be used.

Moreover, Sokrates was really terrible at that time. Someone made their own generators, [the firms S and N] … They took the results from Sokrates, but not from VTT … They could do things based on the Sokrates results, because they had learnt something "too well", and they did not need VTT. The same was actually true with regard to [the CASE tool vendor firm Is].

I still believe that the companies who were involved [in Sokrates] managed to make good use of the results. At VTT, we were always told that this was no tool development project … This was another reason for what happened: they told that it was just a prototype … The generator itself was not a ready-made tool. I can't help wondering why there has not been any active attempt at selling the coding architecture. There aren't any brochures on it available either, for example, indicating that we have software architecture knowledge for sale. I would imagine that it has not been sold in any other way than in projects in which it has for some reason been used. Software architecture knowledge has merely been taken in use in projects by chance, and it has not been viewed as a marketable thing.

Towards the end of the Sokrates project, we had extensive discussions with the companies involved on how to utilise the results. Something had, however, happened in the companies. In many cases, the persons had changed. The structure of their products had changed to such an extent that these things were not topical any more.

On the other hand, since there were no scientific references on the developed code generation techniques, the hopes were only faint for proposing any new European code generation research based on the results. This kind of thinking, from Tekes-funded national projects to EU-funded European projects, was followed at VTT during the early nineties.

No one ever mentioned anything of this.

The original manager of the Sokrates project returned to TKO after his leave of a few months. He helped the project group to prepare for the external evaluation of the project at the end of the Finsoft research program, carried out by national and international experts on behalf of Tekes. The evaluation results, both industrial and scientific, of the project were very good indeed.

There were also some foreigners involved. Their conclusion was that "the research group is on a par with top code generation research in the world". In other words, we were right at the top of code generation research.

However, another evaluation performed by foreign scientific experts on behalf of the Finnish Academy produced results of completely different nature. The evaluators were very unhappy with the lack of any scientific evidence of the principles and novelty of the code generation techniques.

I believe that they did not have anything against code generation, but the design philosophy, because it was based on SA/SD. They did not say anything negative about code generation (you should check this out, if you wish to hold on to your claim).

The researchers wanted TKO to invest money in the commercialisation of the code generator. In connection with this proposal, discussions were conducted if it was right for a national research institute to invest taxpayers' money on developing software tools that would compete against truly commercial CASE tool vendor companies. A decision was, however, made to invest some money on the further development of the generator.

In particular, the Ada language was now replaced by C, as was proposed to the project group already some time earlier. Some brochures and usage manuals were also written, and a licensing policy established. The generator was renamed as Reagenix.

This was not the case. Reagenix is no redesigned Sokrates. It was designed completely anew based on my experience from the pitfalls of Draco, Refine and Sokrates. It was developed in a pre-study, and made ready with surprisingly little effort. From my point of view it was a mistake that it was taken as the continuation of Sokrates and therefore no one was interested in it. On the other hand, no one knew which skills were needed for developing it.

What was worst in this story was the indication that the Reagenix code generator would be the same as Sokrates. My opinion is that Reagenix has been developing slowly here and there, and that there is no clear link between Sokrates and Reagenix.

It is interesting that Reagenix was referred to as some external tool [in the projects carried out by TKO], but nothing was paid for its use. It should have been told that we had this kind of tool available, and some money should have been allocated from the projects to its development.

We had already made our own generator, before the generator of [the firm Is] was introduced, and we had set its price as we thought was appropriate. [the firm Is] launched its product some six months later, and they had a much higher price. In other words, we did not know their price, otherwise we would have used the same price. By the way, Reagenix was introduced in the Technopolis Oulu, before [the firm Is] introduced its own generator. You [Veikko Seppänen] were not there were you? You were in Japan and therefore we could do what we wanted. [The former manager of the Sokrates project] was the section head.

When Reagenix was introduced, [the firm Is] needed to reduce the price of its … code generator to half, which was more than 60 thousand marks, but they had to start to sell it at the price at which we were offering Reagenix, 30 thousand marks. I believe that we really caused business problems for [the firm Is], due to the fact that Finland is such a small country. [The firm] had its own marketing organisation which had to face the problem of everyone asking them about Reagenix, especially since the front end was the same, …, and we had claimed that Reagenix can do everything. Yet, [their generator] was good for generating sequential code only.

Another important extension of the code generator was developed, an operating system model and its prototype implementation. The operating system could be used instead of commercial operating systems as a part of programs produced by the code generator. The idea behind the operating system was patented by the code generation researchers, and it was thus truly original. Some parts of the idea were used in an industrial project carried out for the above-mentioned key customer of TKO. The code generator developers were, however, not involved in this project. The work was performed by another person who was an expert in the customer applications and had conducted several contractual projects for them.

Except that we told [the colleague] how to do it! … It was [his] project. We sold the idea of the single stack operating system. [A person from firm N] was familiar with the idea and supported us – he took part in the steering group of Sokrates.

Investments in the further development of the code generator became a difficult matter after some time, from the viewpoint of the VTT managers, because no considerable contractual or research projects had resulted from the investments.

Yes, the point is that it is not reasonable to think that as early as one year after finishing a [research] project the customers would be there. It may take three years.

The developers of the generator focused, instead of large-scale projects, on small-scale consultation and teaching of the Sokrates-SA method. A new side track of testing was, however, established. The idea was to use parts of the input models used by the code generator for testing embedded programs. This work was a small project done for a private company, and it resulted in yet another tool prototype. The tool was marketed to other companies, but without much success. There were, again, some Finnish and foreign competitors, and the particular testing problem tackled with the tool was perhaps not considered the most serious one by industry.

We first carried out the Sympa project, an analysis of the improvement of their software development practices. Unit testing evolved through that analysis. Once again, it is all the better that small expenses yield the great benefits … as happened in this case, but the organisation [TKO] did not support us in the further development of CUTE. Yet, it was transferred for example to [the firm A], although they had a horrible situation there. Their subcontractor had developed a massive software architecture that needed to be tested before it was taken in use. However, it was not just testing, we also had to deal with the architecture. As the well-known subcontractor and was using a new object-oriented technology, we had to tell them how to test the architecture. Therefore, we just took the consulting role, by commenting what [the firm A] itself was doing. I still believe they got what they wanted, although they could not invest in CUTE as such.

The original developer of the Sokrates code generator aimed at writing his doctoral thesis on the results of the Sokrates project, which would have resulted in a comprehensive scientific evaluation of the basic principles behind the method and the tool. The plan has, however, not yet materialised.

The dissertation would actually not have resulted in this, had it been finished.

We did not have the university background. It [scientific documentation of the code generation principles] should have been done as a part of the Sokrates project. … In my opinion, it was the university attitude [that was missing]. … There was no one to guide us in doing it … [a colleague] had high criteria for publishing, for us they were clearly too high, we should have studied how to describe these things formally.

Another problem was that as we always had two to three customer projects going on at the same time, there was quite simply not enough time for writing any documentation. At the end of the Sokrates project we made a mistake in trying to develop Sokrates [further] without publishing anything at that point. We should have stopped the development during the last year, to write publications. … Later, SA/SD was not any more successful with regard to publishing.

Increasing interest in object-oriented methods also made further investments in an SA/SD code generator more difficult. The focus of research on generic software engineering methods was moving towards these methods instead of the SA/SD method.

At that time, I was following the evolution of object-oriented techniques quite closely. I had already become familiar with these techniques during my diploma thesis work. I was listening to the praising of the techniques by Risto Suitiala [a researcher at VTT Information technology laboratory] and I realised that there was a lot of potential in these techniques. Refine supported some object-oriented features and transformations from these to code. I had discussions with [a colleague at TKO] on how object-orientation and real-time systems would fit together. I was using Object Pascal in my spare time to design a Windows-based user interface program. I even wrote tentative transformation rules from object life-cycle state diagrams to an object-oriented programming language and presented them [to another researcher], who applied them in [a customer project].

The original Sokrates project group had been dissolved, but kept together, in practice, and worked even during their spare time trying to advance code generation techniques. The biggest problem, from their viewpoint, was that the line managers were not willing to invest much money, but were rather looking for larger contractual application projects based on the existing generator, or for an extension of the generator's input language to an object-oriented formalism.

Actually I did not work that much in my spare time, except correcting some errors [in the generator]. As the [research] customers at VTT using the generator did not pay any license fees, these small tasks needed to be done [in my spare time].

The few application projects involving the code generator, lasting a few man-months at most, resulted in a very small return of investment from the viewpoint of the VTT managers.

Who tried to initiate these projects? I was never given any information, although I would have had some ideas and the skills for the job.

The relationships between the line managers and the code generation researchers started, therefore, to cool down. The managers had already been somewhat suspicious, as they had been expecting interaction with other researchers, scientific validation of the results, and a considerable number of continuing projects. The researchers, on the other hand, had been expecting support from the managers for the further development and marketing of the generator.

The relations had cooled down before Reagenix already. [Another code generation researcher] had at that point already lost his interest in project marketing. It had happened several times that he had sold an idea to a company, and then either the representative of [the firm Is] or [the manager of] VTT had come to criticise Reagenix. I did not yet have any appropriate contacts and I had not received any positive feedback on my work [on Reagenix], so I was not interested in marketing. I have to say that for this reason I was not fond of project marketing even later.

It is difficult to find any motivation for marketing, if your ideas are first heavily criticised within VTT. If you succeed in marketing a project, there is always a fear that if something goes wrong, somebody will come and tell you that the failure could have been expected at the very beginning …The [MCS-REA] project was carried out when [the former Sokrates project manager] was working as the section head. Indeed, the … project was the only one that succeeded, from the viewpoint of TKO, and [the former project manager] happened to be the head of the software engineering section then.

The managers of TKO said that this was not good, this was a prototype, a hack-hack, and we could not seriously consider offering it to industry. Now, after being in industry for four years, I have seen what kinds of tools are being used for example in the telecommunication area: the tools developed by ourselves and by universities, public domain tools, and our competitors' tools. I must say that Reagenix was, after all, a very reasonable tool.

The impression that was given by the managers of TKO was that industry was using only top-quality, well-packaged tools, offered by reliable parties. This was the point, VTT did not see itself as a reliable tool developer.

During the Sokrates project, the big generator did not arouse interest in anyone, because it was run on a workstation and was slow, expensive and difficult [to use]. But afterwards, the light-weight [code generator] version Reagenix was produced, which was running on a PC. However, TKO did not want to sell it any more, because [thefirm Is] was seen as a strong competitor. We should have gone abroad, or establish a company. … But the problem with such a company would have been the fact that we had the front end [CASE tool from the firm Is]. … That was a clear problem. In other words, the developer of the graphical front end of our generator had its own competitive product.

During Sokrates, which was a research project, the participating companies got what they wanted. Afterwards, the generator should have been commercialised and taken abroad. In Finland, the big customers had already taken part in the Sokrates project. Otherwise there were only small companies who still could have bought the generator and used it.

By the way, I established several contacts with companies abroad, and in some cases I had quite lengthy interactions with the contacting parties. At the phase, at which an offer was requested and I sent them the information, including the fact that the graphical front end which was needed could be acquired from [the firm Is], the contact would always end.

I guess that they contacted [the firm Is] and asked about Reagenix, and [they]would certainly give their opinion of the matter!

The story does not describe clearly enough that whenever we were trying to take the Reagenix results to the market, the organisation [VTT] would always say no, in one way or another. For example, I was having discussions with [an American CASE tool vendor] concerning the integration of their CASE tool as the front end of Reagenix, and I had also ensured a case example from the Sodankylä observatory, but then [the section head] came to argue with me about me dealing with foreign competitors [of the firm Is], which was not suitable. The co-operation failed, because the managers said that we would not compete against [the firm Is] in this matter. I don't know, [they] perhaps felt that I must be hauled over the coals.

However, what was good about the development of the code generator was that we did implement it, after all. What went wrong was that the generator failed to get sold. I believe that we simply did not know how to sell it, then. There were no previous experiences of this kind of a matter [at VTT]. I would say that we did not have the culture of selling. We were improvising. We, or at least I, did not know what would be the ordinary way for selling such things at VTT. We were just doing our best to invent something, but that was much harder than finding out how to implement the generator.

I implemented the first "big generator" as my diploma thesis. One of the first feelings that came over me when reading the story was a feeling of sorrow, considering the amount of effort that was spent and how bright the people involved were, and how someone else [at VTT] would then be looking for some formal aspects to make this group of people look incompetent, and to create a public opinion of the group not knowing anything about any real problems. Another thing that went wrong was that the SA/SD method was not a very good choice. [The CASE tool of firm Is] was a nice SA/SD drawing tool, but the choice of the SA/SD method was a failure.

We were [also] thinking about establishing a company. But we had, for example [another code generation researcher] and I, already been running a firm for several years. I had been the managing director of my own firm for seven years, [the other researcher] for three years. The firm, as such, was not an interesting idea. We knew what that would have meant. To start a company, we should already have had some customers for the product as well as an established market.

Extending the tools

VTT was reorganised in 1994 to include nine big research units. TKO became a part of VTT Electronics (ELE) and the former section of VTT Electronics laboratory dealing with automation systems a part of VTT Automation (AUT). The researchers of AUT were still interested in the code generator. The basic need of the generator was in rapidly implementing and testing new machine automation algorithms. The generator appeared to be a very effective tool for this purpose, if its users were willing to design the algorithms by using SA/SD. For AUT researchers this was no problem, they had been using the method for several years.

For them, the code generator offered a higher-level, graphical programming language and environment for systematic and rapid production of implementations involving machine control algorithms. Also the line managers of AUT considered the code generator as a very good tool and could not understand, why such an excellent tool had not been utilised in a larger scale at ELE.

But they did not pay anything for it.

The viewpoint of ELE had, however, been to develop and market the tool to professional embedded software developers working in organisations where software design dominated embedded systems development, software design was carried out on a daily basis, and the software needed to be maintained for a long time. This kind of use was different from the needs of, for example, small-scale implementation of new machine control algorithms by researchers focusing on other technologies than embedded software as a means of implementing computerised systems.

I carried out the Kaasu project, of which the customer was very satisfied indeed … We had an article in Tekniikka & Talous, the technical newspaper, about the code generator, and they called us – to me, if I remember it correctly – right away, saying that they would need a control system [to be developed for a gas cromatographer]. My name was mentioned in the article. That must have been why we got the project.

The user gave a profile on how to implement the valve in the equipment, the control system followed the process. They produced a commercial system of the equipment … In that phase, it was a prototype though. They did not have software designers of their own. We did not sell Reagenix, but the control software for the equipment. It was done with Reagenix. An easy job. The schedule held. Nice equipment.

Similar small-scale success stories of the use of the code generator emerged also within ELE, e.g. in connection with the research on fault diagnosis systems. In a few projects, especially in one research project where a computer controlled wood carving machine and its diagnosis system were developed, the code generator was successfully used for implementing both the wood carving algorithm and pieces of the diagnosis software. The needs of this project with regard to the development of embedded software were much similar to the needs of the researchers of AUT.

Tulko is an important reference that is missing in the story. The Tulko demonstration system was built by using Reagenix and the SIC communication protocol. The final demonstration system was a full-scale real-time robot, with a perception system and distributed robotics tasks.

I never got any positive feedback from the small-scale success stories. Yet, at that point the whole [Reagenix] generator consisting of about 5000 lines of Prolog code had been implemented by myself.

Despite the success of these projects, no larger-scale use of the code generator would emerge. One of the main reasons was that there were still no large projects in which the generator could have been applied.

All the time there was the problem that I should have been selling projects, but the only thing that I could do was to go and ask if they had any projects which I could carry out using Reagenix … There was no reason to sell Reagenix to any project. I would not have earned any extra value for developing Reagenix. The projects did not pay anything for the use of Reagenix … We were lacking internal charging on the use of Reagenix … Instead, I have been blamed for having produced hell of a system. I have, however, developed a system that is being used by at least a hundred people … I have also been blamed of always doing everything with Reagenix.

I also see that Reagenix, the technology itself, is not all that beneficial to VTT. We could sell design projects instead, and use it for analysis and simulation. I am just saying that the project portfolio has not been good: the benefits go to the users of the tool, not to the developers.

I will now tell you a typical story. We were at a Hi-tech conference to demonstrate Reagenix. People from [a possible customer company] came to talk to us. We were talking about design methods, generator, etc. and agreed on a visit. I went there with [a former Sokrates member] then, carried out some requirements analysis and summarised what TKO could do for them. They needed, for example, DSP in their system. The real problem for them was the management of the entire product. We proposed Reagenix for this. We had analysed the situation. The problem was to manage the entire product. They were very interested in Reagenix, as it could be used as integrating technology for DSP, communications, etc. We returned to Oulu, told what the customer had said, and [a VTT manager] heard the word "DSP" and sent some DSP project offer there, and that was the end of it. In conclusion, all the things we were managing by ourselves went well, but if the organisation got involved, difficulties would arise.

I have never met a dissatisfied [Reagenix] customer, if the customer sincerely wished to learn how the things were. I have also carried out projects in which Reagenix could not have been used, the Monitori project, for example, in which the system architecture had already been fixed, including DSP and other parts.

The developers of the generator had perhaps also made the other researchers of VTT tired. At a personal level, some people seemed to discriminate anything related to the code generator.

No one ever told me about the success stories. I could never take part in the reporting of the results of the projects [where Reagenix was applied]. I just gave consultation and corrected small errors, if I had some extra time available. It is possible ascribe the fact to the industrial projects related to code generation being small in two ways. The first is that it was small business for VTT, and therefore an unfruitful line of research. The second is that the techniques and skills which were developed were highly efficient. The problems of the customers were solved with little effort and very fast. A good example is the development of the two-phased gas chromatograph [for one of the institutes of VTT]. The work took only little over one man-month.

Our idea was to solve problems and not to begin to profit from the customers. Now we come to the fact that I and [another code generation researcher] had been involved in business for a very long time ... and there were some differing viewpoints. We learnt how to solve the customers' problems and when they learnt that by associating with those guys their problems would be solved fast, they would come again to us in some other matters … In other words, we took a customer-oriented approach … We thought always about the value-for-customer. We have in my opinion taken care of this aspect much better than VTT projects on the average.

Struggling with low interest

Yet another attempt was made to extend the code generator, by investing money in the development of a graphical debugger integrated with the generator. This debugger would have been a novel mechanism, making the generator look even more like a graphical programming environment. Such tools were available only for ordinary programming languages, such as C. The CASE tool sold by the Finnish vendor did not include any debugger. Again, no contacts could be established with this vendor, but instead a small software house was hired to implement the debugger.

It was proposed by [the managing director of the software house] … It looked interesting. Moreover, some people had been complaining about Reagenix not including any graphic debugger. Aniprosa seemed to offer a possibility of carrying out research on graphic debugging.

No internal contacts were made to the group of ELE which had been doing research on the animation of graphical system models for almost ten years.

This is not completely true. We had knowledge of how the other animator was designed. It appeared to be too much tied with a specific SmallTalk/Petri net based implementation. The small software house was used, because it had already implemented an animator for SA/SD diagrams. Politically, this was apparently unwise, because the animator was a direct challenge to the [tool of the firm Is].

The debugger was implemented, but it did not result in any new contractual or research projects where the generator could have been applied.

It was used in one project, though … It was a mistake to allocate the Reanimator work to the [Reagenix] account. I have been developing Reagenix with the money I have earned. A hundred thousand marks was spent on debugging, which was not of any use after all.

On the other hand, no serious attempts were made to extend and modify the code generation techniques for the needs of the rapidly emerging object-oriented software development methods. An example would have been a means of generating C++ programs (the most typical object-oriented programming language) from an object-oriented system modelling language, using the generator and some object-oriented CASE tools.

I would have had ideas, and I was trying to present them occasionally. Yet, no one of the managers of VTT was interested. I even designed a prototype of a generator in a few hours for the object life-cycle method after I had taken a C++ course. This principle was later applied [in a customer project].

Several attempts had been made by the line managers to make the code generation researchers interested in other subjects, so as to ease the situation resulting from the fact that internal investments in the generator could not be continued and that there were no considerable projects in which the generator could have been applied and extended.

I was irritated by the ordinary research results of TKO that did not form any synergetic entity, but were useless when put together – as we would have developed the rear shaft of a Porsche, the gearbox of a Lada, the engine of a Cessna, the body of a Zetor, and so on. Yet, I am satisfied with having expressed honestly what I was going to do. The customary way was to write a proposal using normal research liturgy on how the research would result in new challenges and produce immediately applicable results, and then continue the old line of research. I did not like this kind of humbug. Still, I was able to make the management's idea on my thinking crystal clear.

In one of these "new" projects methods for the implementation of machine vision software was studied, another focused of software component reuse techniques. In both cases, however, the code generation researchers used the projects, in part, to extend the code generation techniques.

If you are referring to the Rekki project, I must say once again that for the most part I did all kinds of tool evaluations and tried to find a reasonable approach to carrying out the work together with the university. Yet, the university was interested only in continuing its own Khoros-related research, which was completely non real-time oriented. We did not do any code generation related research at all in the project. If this [component reuse techniques] refers to the Komppi project, you are presenting just another one-sided viewpoint. I was not able to participate in the literature survey part of the Komppi project. Code generation was not developed further at all in Komppi. I made it very clear, because I was aware of the attitude [of the managers]. I find it very unpleasant, if you use Komppi as an example.

This somewhat irritated the managers, and also some of the other researchers involved in the projects. The results of the projects were accepted by the funding bodies and the industrial partners who were involved, but did not result in any larger-scale use of the code generation techniques.

The American style marketing irritated everyone. They perhaps thought that we would produce some equipment that would free the designer from thinking of anything any more. Instead, we were aiming at defining a language that could be used for solving difficult problems. This was in contrast with what everyone thought. They did not understand … It caused trouble inside TKO, though not outside.

Some code generator licenses, each worth of two man-weeks' contractual development work, had been sold to companies and given for free to some educational institutes. Often this was done as a side job of some development project.

Educational institutes paid modest license fees, VTT institutes used the generator for free ... I still remember how difficult it was for me when I first joined VTT thinking that I could not produce anything. The contribution of investing five thousand in something which would allow industry to make millions was not realised … Our point was that we were offering solutions to customers by which they could produce things cheaply and fast. If the customer could get something at fifty thousand marks, for which they had spent half a million marks earlier, that was only good. This was our way. … And that [kind of projects] could have been sold … One thing is that we were not allowed to sell other than our own work. It was only at the time when I was already leaving VTT when it was told that money could be earned from projects.

Was the line organisation thinking that after investing five millions and it resulting in such small income, it was not sensible to use the developed technique? It could have been developing slowly, after all. A related example is an article published in the Harvard Business Review on the investing and harvesting phases of technologies, as well as the decline. When people read these papers and thought that the whole cycle was five years, General Electric gave up computers, before the decline phase would start. Xerox also gave up computers, although they had object methods and windowing systems [already then].

By the mid-nineties, the original group of code generator researchers had dissolved totally, because the persons were working in several projects not involving code generation techniques. Two of the original developers left VTT in 1996 and 1997, and it seems that this line of research has finished at ELE.

You mean the Sokrates group. I am still here, with all the knowledge. I have sold three licenses during the past six moths, and two other firms have shown interest in it.

I have also organised three courses, each two or three days. I answered to the most recent customer request last week. The total income from industry was about 100 000 marks last year, and the expenses about 30 000 marks. How right were you, after all [when stating that the story was finished]?

Afterwards, the same license plus training was sold to another company, for about 30 thousand marks. Another license was sold separately. They paid my travel expenses, too. In summary, three licenses have been sold after Summer 1997. According to your story, this line of research would be dead! It isn't. The front end is different, it is not [from the firm Is] any more. They wondered [the buyers of the license], why we had not been marketing the tool. I told them that I did not have the time at that point.

From the viewpoint of my short industrial experiences, I would say that VTT has no intrinsic value by itself. It should support the Finnish industry. If the guys have made themselves unnecessary, VTT can be closed and the persons can go and find jobs in industry. Then, a new research organisation can grow if it is noticed that industry does not know enough of some new topic, and the organisation can look five to ten years forward on behalf of industry. VTT is needed for such a purpose, to hold a position at the border of industry, to do what industry itself is not doing. And to help small companies. VTT should not have any intrinsic value by itself.

But for some reason we did not dare to sell this idea of using Reagenix inside TKO. We did everything with traditional methods even inside TKO. In the projects where I was involved, the initial setting was already such that Reagenix could not be used. I always joined the projects in a phase when it was not reasonable to use Reagenix any more … It was not seen at the level where the projects were prepared. It was just the few guys who were involved in Reagenix. For this reason, I should have spotted the customer first, and I should have realised that I needed to sell a certain kind of project to the customer, and then it [the use of Reagenix] may have emerged. Inside TKO, they did not believe in it, they preferred programming from scratch in all projects.

Moreover, if you charge on the basis of working hours, it would not be reasonable to use Reagenix. If the charge had been by the hour and the use of Reagenix had resulted in less hours, the use of Reagenix would not have been reasonable. We made a mistake in the sense that Reagenix was used in many places, in fault diagnosis projects and in the Electronics laboratory, and so on, but we could not get any benefit from the use of Reagenix. The license fee should have been included in the expenses of the projects.

The effort that was saved by using Reagenix could have been used for the further development of Reagenix … Not even licenses were sold, because that was considered an extra. VTT got it for free … Is there even any information of how extensive the use of Reagenix in the projects was, if it is not explicitly mentioned. There were rather many people who became interested in Reagenix.

Reagenix should have been used as part of research projects. And it was used, wasn't it. Another thing is that we should have had Reagenix-based projects. However, if a firm comes to VTT asking about the development of some product, the firm will contact a person who can be found on the list [VTT's service directory, other lists of contact information] and that person actually decides which methods and tools will be used. No one contacts me. I do not have any contacts myself.

Summary

There are certain key technical and human reasons for the developments of code generation activities that took place at VTT, which are closely related to each other. From the technical viewpoint, the developed code generation solutions seem to have stuck with the SA/SD method, when industry was already gradually taking object-oriented methods in use.

Did it really took object-oriented methods in use? After all, the SDL method with code generation features has become increasingly popular, and it resembles SA/SD … One can also think that the five million that were invested were for the need of industry. What if the investment had not been made? The good point that was not told in the story is that the SA/SD method was brought to industry. [One of the code generation researchers] has a remarkable role in that. SA/SD is perhaps the first real design method in the area of embedded systems. [The researcher] is among the first persons who giving training on this method in Finland. Earlier, there had only been foreigners ... At present, when I talk to anyone out there, they do not know me and they are saying that they use SA/SD. When I ask why they are using it, they will tell me that VTT has brought SA/SD further … Yes, indeed … Some of those persons had taken my course.

Object-oriented methods emerged rapidly, but no serious attempts were made to link the code generation expertise to them.

There is a clear difference between Reagenix and the object-oriented world. Reagenix takes care of real time, concurrent control and supervision sequences and asynchronous communication between them. Object-oriented methods, in contrast, can be used to take care of traditional computing and management of stored data … The group [of code generation researchers] did not omit object-oriented methods due to ignorance or arrogance. The concepts were familiar [to the group] already from the computer science journals of the beginning of the eighties … The possibilities and problems were known.

Yet, the developed solutions were very flexible with regard to applications, target hardware and even CASE tools. Although the code generation functions that were developed were hard coded inside the code generator, their principles were made explicit by coding rules.

Why were there no serious attempts? I was never even dropped a hint of this kind of possibility. Usually anything related to code generation was flatly knocked out as Reagenix bullshit. My experience was hardly ever used. On one occasion, I may have made a one week evaluation on the generation of code from Statecharts models.

From the viewpoint of networking, the isolation of the code generator researchers both internally at VTT and externally from the key customers proved to be a problem.

Yes, indeed, from outside it looked outside like a homogenous group that did exactly the same things. By the way, [a colleague] always said that whatever he asked from anyone of us, he always got the same answer. We had thus the same understanding, although we did not write it down, perhaps could not write it down … It was a well functioning unofficial organisation! … In the MCS-REA project, for example, we had very difficult problems that were solved by means of brainstorming.

Moreover, disagreements arose with some technology experts, such as hardware designers, who may have benefited from co-operation. Yet, several successful usage cases of the code generator can also be identified. Most notably, the tool proved promising for the prototyping needs of other than industrial software professionals, for the needs of researchers, actually! Networking with these people, and perhaps with commercial tools vendors that they used, might have resulted in some kind of commercialisation of the code generator technology. Internally, a number of persons adopted a negative or at least suspicious attitude to the code generation research.

The message that the idea of the code generator with all its supporting systems being capable of solving "all problems" was particularly irritating. One could speculate whether a similar development had taken place in industry, if the developers of the generator had succeeded in marketing it to a greater number of customers.

Why wasn't any bigger business created around code generation? VTT did never come to an agreement with itself upon what it wanted from code generation. The lines of continuing work that were brought down include:

- *Co-operation with the American tool vendor – could have resulted in a market worth of millions; and*

- *Development of code architecture for the 8051 processor – would have resulted in a market of hundreds of thousands of marks, because Finland was full of machine firms that were not interested in programming operating systems calls.*

There were a lot of people showing interest at [automation] fairs. The idea was to investigate the design phases that preceded coding. The core competence is there, but it can not be seen, in other words, the knowledge of how to solve concurrence problems … and the development method, analysis and so on.

It seems that those industrial practitioners who actually used the generator or the code generation rules sought for solving the software development problem by focusing on, for example, automation design and trying to make programming a rather mechanical task that could be carried out with a tool. For them, the message mentioned above was promising. For people who where developing software as their profession, the message did not go through.

Why real software professionals not interested, but just machine automation people? Did portrait painters become enthusiastic about cameras in the late eighteen hundreds?

The developed code generation technique seems at the first sight to make some central software engineering skills obsolete … I do not know, if software professionals became interested in the first Fortran compilers, because it made obsolete a great deal of central skills [in the fifties], but at the same time it made it easy for mathematicians to program algorithms by themselves.

I myself took even greater interest in code generation than many others, because for seven years I had been risking my own money in tasks contracted at a fixed price. It always paid to think thoroughly how to do the job at the smallest effort. One very satisfied customer was [the firm Se]. How long do you believe that the development of the software package for their product took?

It took only 17 hours! I did a demonstration system during two nights in a hotel, then I showed it to [them] and asked, if it would be embedded in the target hardware. We spent five hours on a Friday evening doing that, then … Software developers still have many problems. These [Sokrates/Reagenix] things have not become obsolete at all.

During one course, a technician said to me that when he had told his colleagues about this method [Sokrates-SA] and the generator, they had really become frightened and thought that if these kinds of tools were used, they would lose their jobs … They could not see that they would then be able to focus on higher-level issues, new designs and so on.

I would say that this is a rather non-democratic issue. There were smart guys [in the firm Nm]. Things can be non-democratic … It was an appropriately sized group. The chief designers were programming themselves. In industry, some people just work from eight to four and do not want to learn new things. They use what they already have learnt. Other groups may realise that they can utilise some new things … The people at [the firm Nm] were very smart indeed … And they were very busy, too … They were not interested in coding, but in getting their equipment ready. They did not get any kick out of programming, but of the fact that the equipment was completed … They saw their job not as coding, but as implementing the equipment.

Published by

Vuorimiehentie 5, P.O.Box 2000, FIN-02044 VTT, Finland
Phone internat. +358 9 4561
Fax +358 9 456 4374

Series title, number and report code of publication

VTT Publications 418
VTT–PUBS–418

Author(s)
Seppänen, Veikko

Title

Competence change in contract R&D
Analysis of project nets

Abstract

This research addresses the problem of what to consider as competence in research and development (R&D) and how does it change in contractual projects. Two cases are investigated, so called fault diagnosis systems and code generation methods and tools. They involve the Technical Research Centre of Finland (VTT) as a supplier and several companies as customers. Projects established by these parties from 1985 to 1998 have been analysed, to study how the supplier's competence evolves. The research is based on the analysis of several thousand pages of documents and interviews of some fifty persons. The change of competence is evaluated and explained within project-based relationships called project nets. Differing objectives and goals of the interacting parties are found to greatly affect project nets, and thereby the evolution of the supplier's competence.

In the code generation case, the focal VTT managers aimed at creating a versatile portfolio of fully contractual projects, for machine automation applications in particular. The approach did not work as intended. The code generation researchers produced design methods and tools that were, in the end, mostly utilised inside VTT – which neither benefited the contractual R&D business of VTT, nor resulted in explicit core competence. In the fault diagnosis case contractual projects were, however, used to create what can be considered as core competence of VTT as an R&D supplier.

Still, even the code generation competence survived through many years and conflicting viewpoints to form a basis of business – not for VTT, but for some of the original inventors. This indicates both the key role of individuals in R&D and the need for strategic management of evolving competence. Development of functional capabilities to solve problems in certain applications is found in this research to be essential for the building of core competence of an R&D supplier. Flexible use of generic engineering techniques and implementation technologies is also important. However, this depends on how skilled customers are in this regard.

The final part of the dissertation is devoted to the question of how to manage competence, based on the empirical insights of the research. The goal is to help pave the way for strategic relationship-based competence management in contractual R&D.

Keywords
core competence, customer relationships, R&D management, industrial networks, software engineering, fault diagnosis, code generation, electronics industry

Activity unit
VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland

ISBN			Project number
951–38–5577–5 (soft back ed.) 951–38–5578–3 (URL: http://www.inf.vtt.fi/pdf)			
Date September 2000	Language English	Pages 226 p. + app. 29 p.	Price F
Name of project		Commissioned by	
Series title and ISSN VTT Publications 1235–0621 (soft back ed.) 1455–0849 (URL: http://www.inf.vtt.fi/pdf)		Sold by VTT Information Service P.O.Box 2000, FIN–02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374	