

Teemu Tommila, Juhani Hirvonen & Antti Pakonen

Fuzzy ontologies for retrieval of industrial knowledge – a case study

ISBN 978-951-38-7494-0 (URL: <http://www.vtt.fi/publications/index.jsp>)
ISSN 1459-7683 (URL: <http://www.vtt.fi/publications/index.jsp>)

Copyright © VTT 2010

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 5, PL 1000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 5, PB 1000, 02044 VTT
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O. Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 4374



Series title, number and
report code of publication

VTT Working Papers 153
VTT-WORK-153

Author(s) Teemu Tommila, Juhani Hirvonen & Antti Pakonen		
Title Fuzzy ontologies for retrieval of industrial knowledge – a case study		
Abstract <p>Information and knowledge are critical resources in operating and maintaining industrial process plants. The basis for the required knowledge repository is created at the design stage and extended after the hand-over of design data with modifications, maintenance histories and operational experiences. It is in the common interest of plant owners, engineering contractors and equipment and material suppliers to manage, communicate and utilise the information over the whole life-cycle of the plant. Our goal was to better “mobilise” knowledge stored in heterogeneous databases to users with various backgrounds, geographical locations and situations. The working hypothesis of the research was that fuzzy mathematics combined with domain-specific data models, in other words, fuzzy ontologies, would help manage the uncertainty in finding information that matches the user’s needs. In this way, this paper places itself in the domain of knowledge management. The main goals of the report are to give practical examples of fuzzy ontology in an industrial context, show how such ontologies can be developed, to test the functionality of the applied formalisms and tools, demonstrate the feasibility of fuzzy ontology in searching information from a knowledge base, and describe the next development tasks.</p>		
ISBN 978-951-38-7494-0 (URL: http://www.vtt.fi/publications/index.jsp)		
Series title and ISSN VTT Working Papers 1459-7683 (URL: http://www.vtt.fi/publications/index.jsp)		Project number
Date November 2010	Language English	Pages 54 p. + app. 2 p.
Name of project	Commissioned by Tekes, Metso Automation, Kemira, Rautaruuki, UPM	
Keywords Knowledge management, fuzzy ontology, uncertainty	Publisher VTT Technical Research Centre of Finland P.O. Box 1000, FI-02044 VTT, Finland Phone internat. +358 20 722 4520 Fax +358 20 722 4374	

Preface

This report is an outcome of a joint research project KNOWMOBILE (Knowledge mobilization) carried out by the Institute for Advanced Management Systems Research (IAMSR) of Åbo University, and VTT Technical Research Centre of Finland. The research was funded by Tekes (Finnish Funding Agency for Technology and Innovation) and the industrial companies Metso Automation, Kemira, Rautaruukki, and UPM. The authors wish to especially thank the representatives of our industrial partners who have taken the time to provide us with data and valuable feedback.

Contents

Preface	5
1. Introduction	7
2. On fuzzy ontologies.....	9
3. Goal – a semantic tool for retrieving operational knowledge	13
4. A fuzzy ontology for process industry	16
4.1 Basic domain concepts	16
4.2 Modelling the inexactness.....	20
4.3 Presenting the keyword ontology	22
4.4 Principles of fuzzy queries	25
4.5 Event types	27
4.6 Systems at an industrial plant	29
4.7 Plant functions and activities.....	31
4.8 Process variables.....	33
4.9 Materials and substances	34
5. A demonstration application.....	36
5.1 Demo architecture and implementation	36
5.2 Demo user interface.....	38
5.3 Extension to an external information source	42
5.4 Evaluation of the demonstration.....	45
5.5 Ideas for further development	47
6. Summary and conclusions	50
References	53

Appendices

Appendix A: Publications of the KNOWMOBILE project

1. Introduction

Information and knowledge are critical resources in operating and maintaining industrial process plants. The basis for the required knowledge repository is created at the design stage and further extended after the hand-over of design data with modifications, maintenance histories and operational experiences. It is the common interest of plant owners, engineering contractors and equipment and material suppliers to manage, communicate and utilise the information over the whole life-cycle of the plant.

Today, information and communication technologies are a necessity in information and knowledge management. Computer aided design tools, standardised data models and exchange formats enable the creation and distribution of information in electronic format. Despite a major portion of life-cycle information is communicated and stored as unstructured documents, like technical specifications, meeting notes and e-mails. Differences in terminologies, representation standards and ways of thinking make knowledge management a challenging task.

Today's information engineering sees unambiguous and detailed information models as a necessary, though not sufficient, foundation of all information systems. Data models, often defined in domain-specific standards, are presumed to perfectly represent the contents and meanings of the real world. In the future, shared conceptual models, *ontologies*, are believed to integrate all information sources and types in various life-cycle stages.

In practice however, ontologies and their integration will be far from complete. First of all, human thinking and communication is inexact, person-dependent and thus error-prone. Secondly, detailed data models become very large and hard to maintain. Some part of the information is inevitably unstructured and must be processed by humans or computers with capabilities in natural language processing. In particular, existing unstructured documentation must be also made useful in the future. Finally, ontologies are specific to certain application domains and purposes. This creates the problem of mapping various ontologies to each other. In most cases, exact correspondence of concepts does not exist.

Even if strict and extensive ontologies should be the goal in many applications, for example in the engineering domain, it can be concluded that uncertainty is here to stay. This adds to the challenges of information and knowledge management. Dependable reasoning, analysis and transformation of data with computers have their limits. Beyond that, approximation and human expertise are required both to create and to search useful knowledge in a specific context.

This report is an outcome of the KNOWMOBILE project (Knowledge Mobilisation), a joint effort by Åbo Akademi University and VTT. Its goal was to better "mobilise" knowledge stored in heterogeneous databases for users with various backgrounds, geographical locations and situations. The working hypothesis of the project was that fuzzy mathematics combined with domain-specific data models, in other words, *fuzzy ontologies*, would help manage the uncertainty in finding information that matches the user's needs. In this way, KNOWMOBILE places itself in the domain of knowledge management. Traditionally, it has used techniques like text and natural language processing, data mining and metadata annotations. Recently, increasing interest has been directed towards ontology-based information retrieval (Fensel 2002, An et al. 2008). Fuzzy ontologies have been suggested by several researchers (Parry 2006, Bordogna & Pasi 2000, Widyantoro & Yen 2001, Sanchez & Yamanoi 2006, Gallova 2007).

The specific purpose of this paper is to describe an industrial demonstration of fuzzy ontologies in information retrieval. The application is taken from the paper industry where problem solving reports are annotated with keywords and then stored in a database for later use. Our claim is that fuzzy keywords provide benefits in searching for problem reports and other operational experiences from a knowledge base. The main goals of the report are:

- give practical examples of fuzzy ontology in an industrial context
- show how such ontologies can be developed
- test the functionality of the applied formalisms and tools
- demonstrate the feasibility of fuzzy ontology in searching information from a knowledge base
- describe the next development tasks and demonstrations.

The rest of this document is organised as follows. Chapter 2 gives a short introduction to fuzzy ontologies. Then chapter 3 defines the goal of our demonstration – a knowledge base of event reports annotated with fuzzy keywords that allow intelligent queries. Furthermore, this chapter explains the application domain, scope and viewpoint selected for the demonstration. Next, the purpose of chapter 4 is to define a set of domain-specific keywords as seen by the users of the knowledge base. It also discusses the principles of reasoning on the keyword ontology, i.e. finding the closest neighbours of a given query term. Chapter 5 describes the demonstration application. Finally, chapter 6 concludes the report with the lessons learned. A list of publications produced during the KNOWMOBILE project can be found in Appendix A.

2. On fuzzy ontologies

While having its roots in philosophy, the term ontology is today popular also in computer science. In general terms, an ontology is an explicit formal specification of a shared domain conceptualisation – the objects and concepts of the domain, and the relationships that exist between them (Gruber 1993). Ontology can be used to describe a domain and to reason about it. It is often developed for a specific purpose and application area. Large information systems often need to combine several domain ontologies. This is a major challenge since separately developed ontologies are usually incompatible. Upper ontologies have therefore been defined that apply to a wide range of domains and can be used to integrate dedicated ontologies (e.g. Salim et al. 2004, Batres et al. 2007).

Typically ontologies describe concepts (classes), individuals, properties and relations relevant for an application area. In addition to conceptual models, ontologies apply formal Description Logics (DL) to define the rules and axioms that describe the logical inferences and fundamental truths of the domain. An ontology language, such as the Web Ontology Language (OWL, see W3C 2009), is used to encode the ontology. As such, an ontology is far more complicated than a taxonomy that is mostly focussed on the classification of concepts in a domain.

Ontologies allow the semantics of a domain to be expressed in a language understood by computers, enabling automatic processing of the meaning of shared information. Ontologies are a key element in the Semantic Web, an effort to make information on the Internet more accessible to agents and other software. Search functions based on ontology languages such as OWL gain intelligence by relying on conceptual, semantic links instead of simple string matching. Accordingly, ontologies have gained a central role in many knowledge management tools, especially for efficient integration of knowledge from different sources. Applications have been proposed in the field of industrial automation as well (e.g. Pakonen et al. 2007, Viinikkala et al. 2006, Obitko & Marik 2003).

Semantic Web has been criticised for not addressing uncertainty. Ontologies are crisp, whereas in the real world, relations between objects often include aspects that cannot be expressed in crisp logic (Holi & Hyvönen 2005). The world, and certainly also the at-

tempts at formalising it, are full of contradictions, and traditional, monolithic knowledge bases have major problems with inconsistency (Thomas & Sheth 2006). It seems that the reason uncertainty has not been included into the backbone of the Semantic Web is that the formalisms lack scalability. However, the Internet has been successful specifically because it does *not* expect all authors and producers of documents to follow a strict set of standards in terms of labelling, indexing, and structure (Parry 2006). An ontology that perfectly captures a person's understanding of the world is of little use for someone with a different view (Parry 2004).

Fuzzy ontologies have been proposed as a solution for addressing semantic meaning in an uncertain and inconsistent world. As with fuzzy logic, reasoning is approximate rather than precise. The aim is to avoid the theoretic pitfalls of monolithic ontologies, facilitate interoperability between different and independent ontologies (Cross 2004), and provide flexible information retrieval capabilities (Widyandoro & Yen 2001, Thomas & Sheth 2006).

The background is in fuzzy logic and fuzzy set theory. To get the idea, let's look at a domain, say real numbers, integers between 0 and 100, or discrete entities such as paper machines in the world. In traditional set theory, an element either belongs to a set or not. In the case of a fuzzy set this is not black and white. Instead, an element belongs to a set with a certain *membership degree* between 0.0 and 1.0. For example, a 28 year old person might be considered a "young person" to the degree of 0.8. In fuzzy set theory, a fuzzy set is defined by its membership function, usually denoted by μ , mapping each element of the domain to a membership degree value (Figure 1). A *fuzzy number* is a fuzzy set of numerical values like real numbers or integers. It does not refer to a single value but to a distribution of values. For example, the fuzzy number 0.8 could be something between 0.7 and 0.9 with a peak value of its membership function at 0.8. Using membership functions, the basic set-theoretic operations, e.g. union, intersection and complement, can be defined in various ways. For example, the intersection $A \cap B$ of two fuzzy sets A and B can be derived by selecting the minimum of their membership functions as illustrated in Figure 1. On the other hand, the union $A \cup B$ can be defined as the maximum of the membership functions.

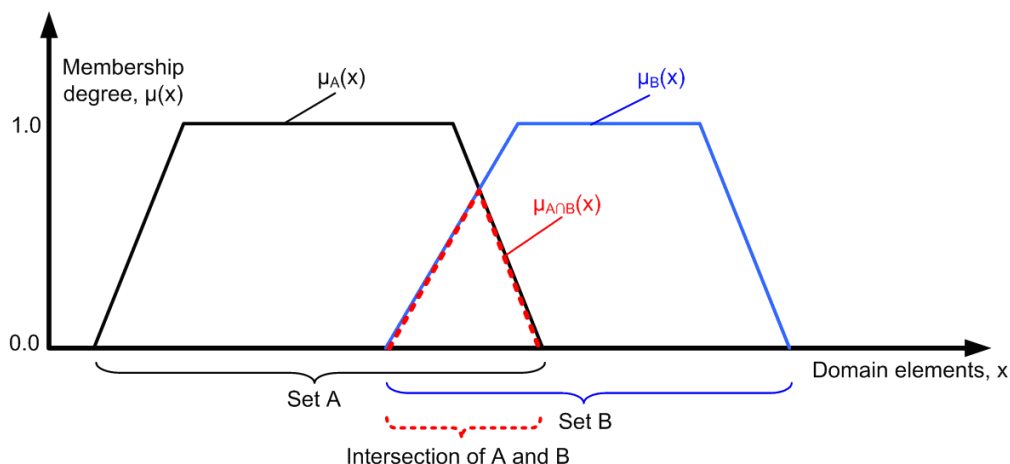


Figure 1. Fuzzy sets are defined by their membership functions.

A term in a fuzzy ontology can have many different meanings, each with an assigned membership value. The fuzzy mapping makes it possible to retrieve information from systems with inconsistent, even conflicting notions on domain vocabulary (Thomas & Sheth 2006). Fuzzy ontology based information retrieval enables great freedom in representing not just the document content, but also the information needs (Bordogna & Pasi 2000). A practical application of such flexibility is the extension on information queries. A search engine utilising a fuzzy ontology can extend a query to also cover related query terms (from different ontologies) that likely have similar meaning to those selected by the user (Thomas & Sheth 2006, Widyandoro & Yen 2001).

Fuzzy ontology definitions found in literature are, quite naturally, influenced by both fuzzy set theory and fuzzy logic, and existing ontology languages (e.g. Parry 2006, Widyandoro & Yen 2001, Sanchez & Yamanoi 2006). As an example, one author, David Parry (2006), bases fuzzy ontology on the concept that each term or object is related to every other term in the ontology, with a degree of membership μ , where $0 < \mu < 1$; μ corresponds to labels like "slightly", "partially", "strongly". For each term;

$$\sum_{i=1}^n \mu_i = 1$$

Where n is the number of relations a particular term has, and $n = (N - 1)$, with N representing the total number of ontology terms. For each relationship, the membership value is not necessarily commutative. The membership value from term A to B, μ_{AB} , may be different to the value from B to A, μ_{BA} .

Parry (2006) provides an example using the term "Apple". In the ontology scheme depicted in Figure 2, an "Apple" can be understood e.g. both as a fruit and a computer company. Different membership values can be applied to each possible interpretation, depending on the context of the user. The ability to handle such ambiguity is useful for a generic search engine.

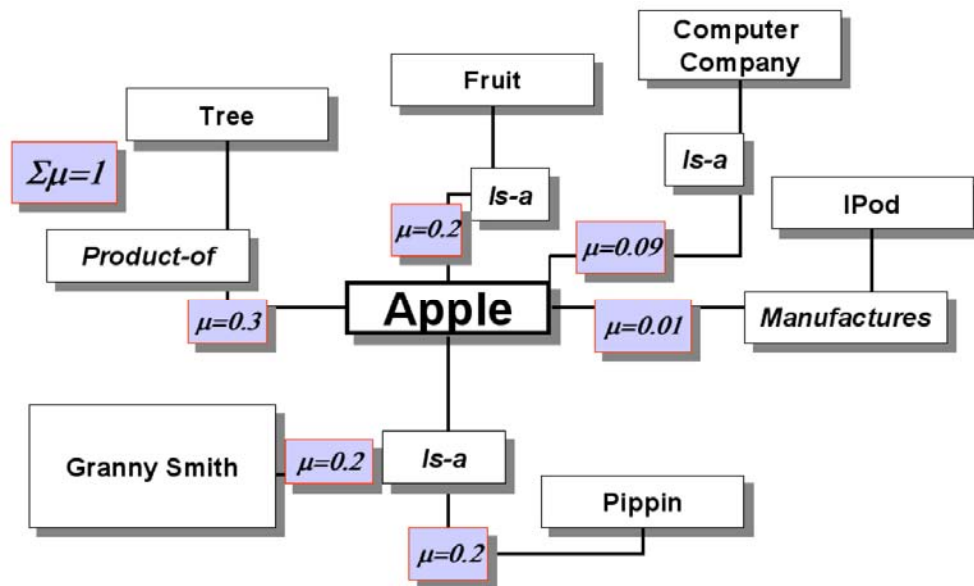


Figure 2. A fuzzy ontology scheme example (Parry 2006).

In the literature, many formalisations for introducing fuzzy constructs to conceptual models can be found. There is, however, no universally adopted approach, nor are there mature software tools. In our research we have taken a practical approach. We began by looking at our industrial domain from the point of view of an intelligent search application, and defined the kinds of ontology constructs that we deemed necessary to effectively capture the domain semantics. We then developed a way to work with fuzzy relationships and began to experiment with the constructs in a demonstration application. The formal definitions of a fuzzy ontology are given by Carlsson et al. (2010).

3. Goal – a semantic tool for retrieving operational knowledge

With the increase in the degree of automation and the amount of instrumentation and ICT systems at industrial process plants, the role of humans (of which there are fewer than before) has shifted from simple monitoring to more knowledge-based tasks – supervision and decision-making regarding ever larger and more complex parts of controlled plants. As a result, tools for exchanging expert knowledge have become integral parts of control systems (Paunonen et al. 2006). Thanks to standardised models for plant information (e.g. ISO 15926 2004 or OPC Unified Architecture, www.opcfoundation.org/ua), information flow between different plant systems is no longer an issue. Indeed, knowledge in the form of written reports is stored in different systems, such as electronic diaries, maintenance databases, or laboratory information management systems. Attachments like trend graphs and video files can be easily included in the reports.

With existing technologies, gathering and storing knowledge is relatively easy. However, efficient retrieval of it is often a challenge. While applications such as electronic diaries are supposed to act as "organisational memory", retrieval of such knowledge is often based on rather simple text applications. Intelligent methods like case-based reasoning have been proposed, but those tend to emphasise measurement data over written reports. Conceptual models have been created to facilitate keyword-based annotation and search, but several process operators have stated their lack of motivation for writing properly detailed, structured, and annotated reports, since their experience has shown that finding anything of worth from the systems is hard.

In the KNOWMOBILE project, we developed a concept of a tool for searching plant knowledge with a search engine based on a fuzzy ontology. The usage scenario for the tool was that a process expert, dealing with a problem in the process chemistry of a paper machine, wishes to find past problem solving cases of a similar setting in order to find possible solutions to a current issue.

This setting is a universal one: pieces of knowledge, called "nuggets", are written and stored by companies on different domains in the form of incident reports, models, recommendations, etc. Our viewpoint is deliberately narrow, as even within the domain of

a paper mill there are other stakeholders besides chemistry experts; for example, the plant owner and equipment suppliers, who are also interested in the performance of the plant. While trying to preserve some general applicability, we have focused on the chemistry of the "wet end" in order to limit the work effort needed to construct the domain ontology and concentrate on a subject on which domain expertise and actual data were available.

Nuggets are documents that can contain all kinds of raw data or multimedia extracted from different information systems. An expert author annotates the nuggets with suitable keywords, and it is these keywords that the search is then based on. In addition to providing exact results to queries, the tool uses a fuzzy domain ontology to extend the query to related keywords (Figure 3). As a result, the search results include nuggets that may not necessarily deal with exactly the same process equipment, variable, function or chemical, but nuggets that may still provide valuable insight to solving the problem at hand.

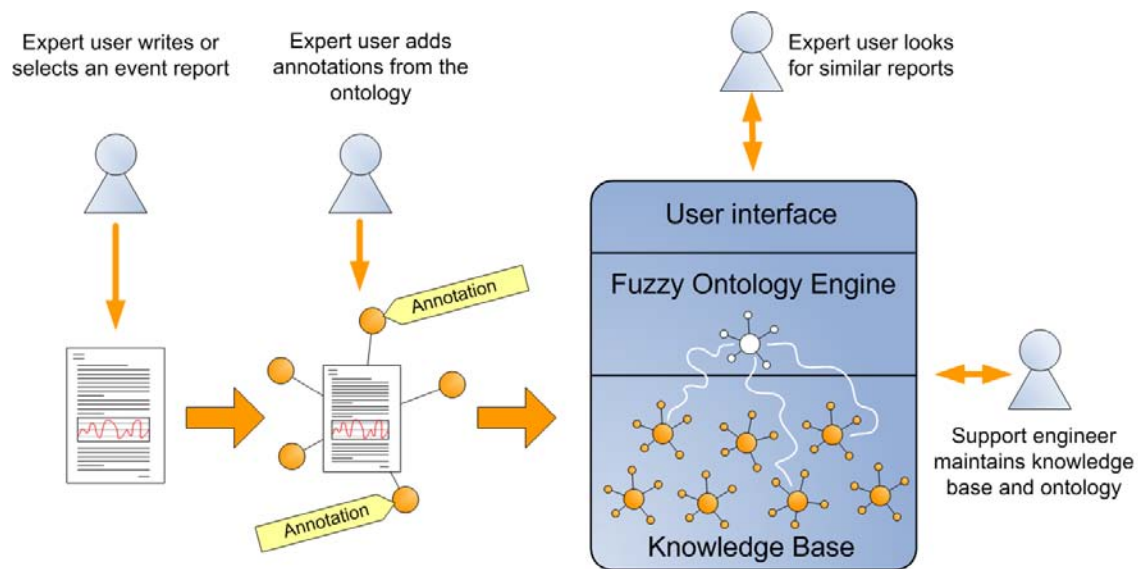


Figure 3. System concept – a knowledge base of event reports that are annotated with domain-specific keywords.

The use of an ontology helps make sure that the extended keywords actually have a common sense relation to the concepts used in the query. For example, the process section "wet end" is related to the function of "forming" of the paper web, so a nugget describing forming-related activities might be relevant if the user is interested in the wet end. Similarly, since the process component "head box" is a physical part of "wet end", similar conclusions can be made. Applying a fuzzy ontology makes it easier to use some flexibility in the query. When strict queries come up with too few results, extension of the query to find also related nuggets can be quite useful.

3. Goal – a semantic tool for retrieving operational knowledge

It is clear that the terms defined in the ontology should be familiar to the users of the system, e.g. for plant operators, process developers and chemical suppliers. On the other hand, the terminology should be based on sound principles of conceptual modelling and ontology development. Furthermore, the goal should be to have common concepts and plant models with equipment manufacturers and engineering contractors that provide the initial information and often participate in modifications and upgrade projects. This creates a link to the ongoing development of engineering data models for various industrial areas. Fuzzy keyword ontologies should thus make use of relevant product and plant modelling standards and more general upper ontologies.

The idea of a query based on keywords is nothing new, and we do not presume that a fuzzy ontology based query mechanism would be the ultimate solution to knowledge retrieval. Rather, we expect that fuzzy ontologies could serve along with other techniques (those based on "traditional" ontologies) as an alternative means. Ignoring the contents of the nuggets in the search is not a strategy we support, but merely a current simplification that allows us to focus on our key research question: How to build fuzzy ontologies for the process industry domain to enhance knowledge retrieval?

4. A fuzzy ontology for process industry

As described in the previous chapter, the KNOWMOBILE project evaluated the applicability of fuzzy ontologies in the context of information retrieval based on domain-specific keywords. Different from the usual thesaurus approach, the keywords available to the user formed a fuzzy ontology. The goal of this chapter is to describe the "fuzzy keywords system" from an end-user's point of view.

4.1 Basic domain concepts

Our demonstration works with both engineering and operational knowledge of an industrial plant. Therefore, the fuzzy ontology should not be developed in separation from existing engineering tools and knowledge repositories, but existing terminologies, taxonomies and data models should be used if possible. This leads to a taxonomic system consisting of several layers as illustrated in Figure 4:

- Top layer: general concepts (i.e. based on international standards) that apply to several industries.
- Middle layer: vocabulary defined and shared by business partners (within a certain industry, again based on standards) to share knowledge of, e.g. the type and structure of process equipment. This layer extends the top layer with domain-specific keywords.
- Bottom layer: custom, company-specific concepts, e.g. specific products and component types, or even individual process plants.

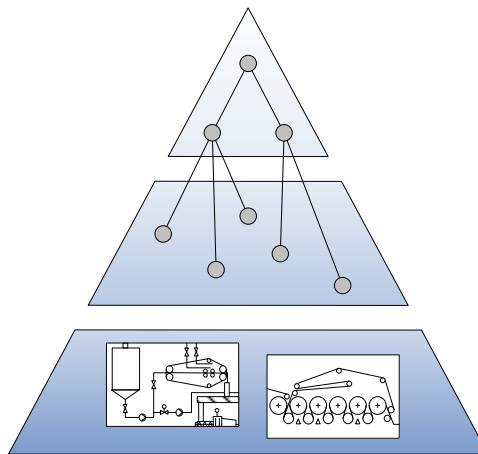


Figure 4. Taxonomic layers – Generic concepts, domain-specific keywords and individual process plants.

In order to speak about an ontology, our "system of keywords" should represent concepts, properties, relationships, axioms, and reasoning schemes relevant for the application area. On the basis of various upper ontologies and industrial data models we identified that the following keyword categories are needed to characterise event reports:

- Systems: types of real-world components of a process plant, e.g. machines, buildings, software and people.
- Functions: phenomena and activities carried out at an industrial plant in order to fulfil its purpose.
- Variables: properties and state variables of various entities, e.g. temperature and paper grammage.
- Events: types of interesting periods of plant life described in event reports, e.g. test runs or equipment failures.
- Materials: raw materials, products, consumables etc. handled in a process plant.

Our basic approach to conceptualise our application is shown in the informal UML class diagram below (Figure 5). Event reports describe events that are related to various entities of a process plant, e.g. to equipment, processing functions and materials. Nothing is assumed about the internal structure of event reports. Instead, they are characterised by an expert with keywords selected from a fuzzy ontology. The expert can select the keywords from five categories: event, system, function, material and variable. All keywords represent an entity type and can have subtypes and smaller parts. In the KNOWMOBILE demonstration tool (see chapter 5) keywords are used to characterise event reports and other nuggets stored in a database. Therefore, keywords can be under-

stood as representatives of populations of real-world entities that overlap and are related in many ways. For example, the keyword "paper machine" (cf. Figure 6) might represent the set of all paper machines in the world.

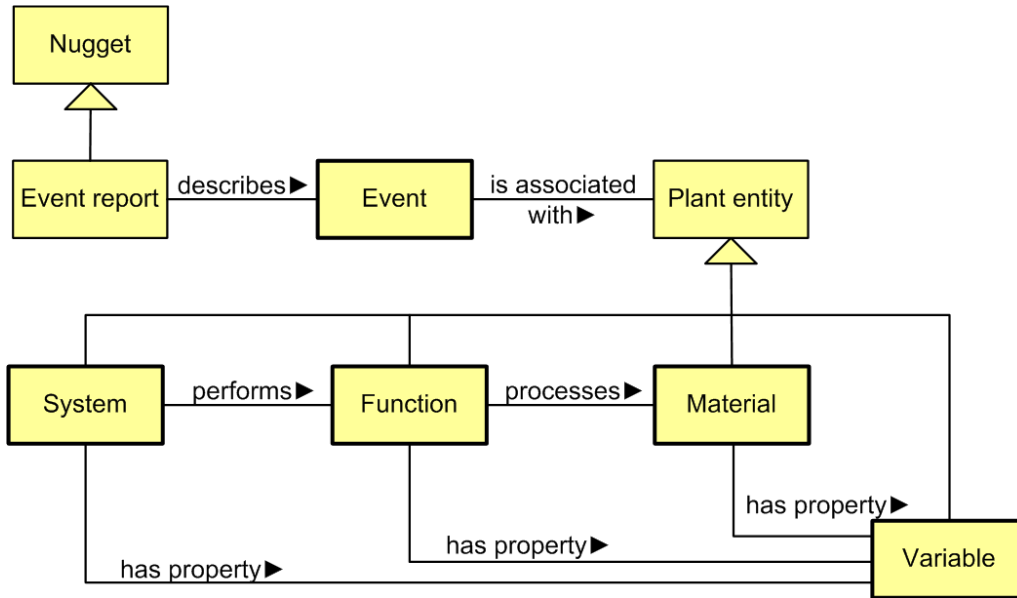


Figure 5. Overall domain concepts.

Classification (*is-a*) and decomposition (*part-of*) can be found in most ontologies and data models. They are important in the industrial context as well. So, the keywords in each category are linked by *is-a* and *part-of* relationships as illustrated in Figure 6. Furthermore, the ontology should model functional and other kinds of dependencies between keywords in various keyword categories. As an example, *systems* can be or are used for some purposes, i.e. they play various roles in carrying out one or more *functions*. This creates a link between the keywords "wire section (a part of paper machine)" and "formation (a quality measure of the produced paper)". Modelling classifications, decompositions and various dependencies leads to a situation where we have a taxonomy tree for each keyword category and a set of partonomy (part-of relationships) trees describing the decomposition to various domain entities (Figure 7). In addition, there are dependency relationships linking keywords to each other.

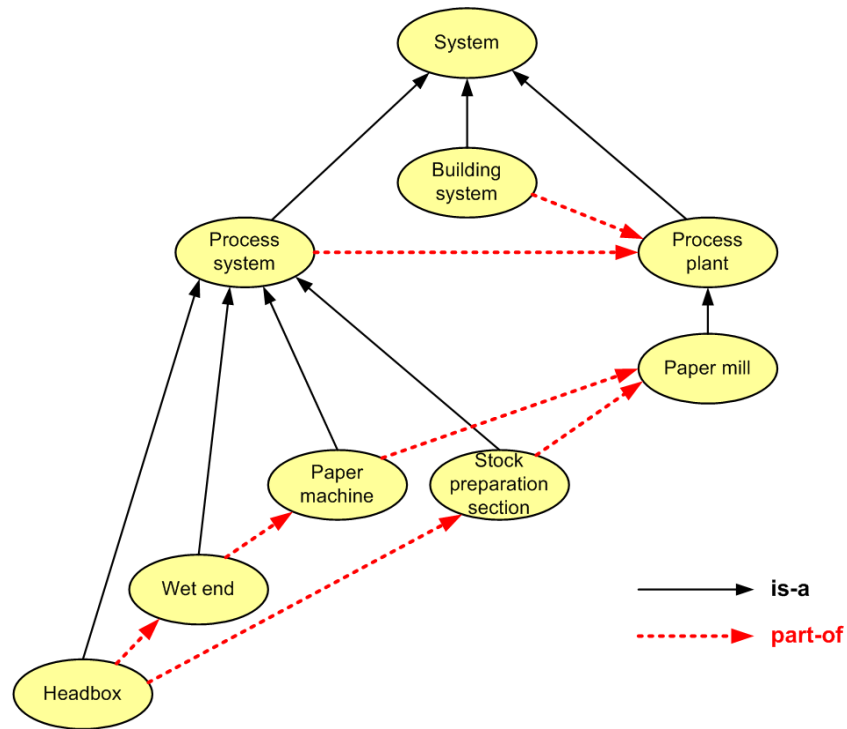


Figure 6. Examples of is-a and part-of relationships in the system category.

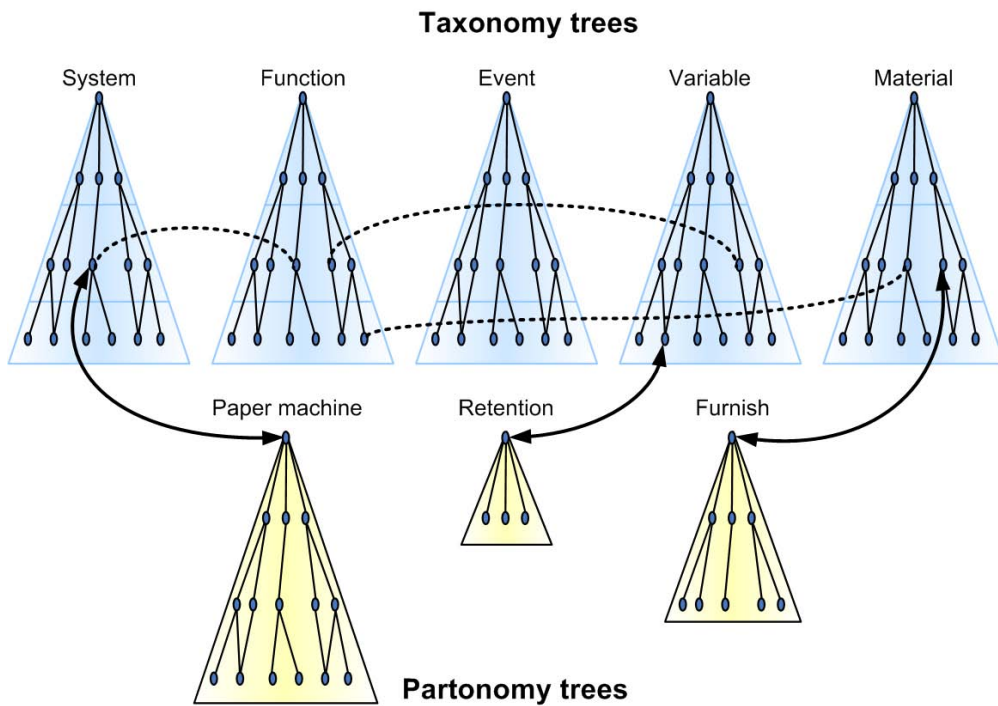


Figure 7. The fuzzy ontology defines classification, decomposition and miscellaneous dependency relationships between keywords.

4.2 Modelling the inexactness

Even a crisp ontology might help us find keywords related to a given search term. For example, subclasses and parts of a domain entity can be used to extend the query given by a user. Furthermore, a machine typically used to carry out a certain function is a candidate for searching related event reports. However, all links between keywords are not equally important to the user. First, the sets of real-world entities represented by the keywords have different degrees of overlap. Second, expert users also tend to give different meanings for the words they use. So, we need a way to model this imprecision and to prioritise search results according to it. Fuzzy mathematics is the approach selected in this report.

This imprecision can be introduced in several ways. For example, fuzzy relationships and property values can be attached to the ontology. Another approach is to use fuzzy mappings from symbols used in human-to-human communication to the concepts in a crisp ontology. Like in every-day language, words can have more than one meaning as illustrated in Figure 8).

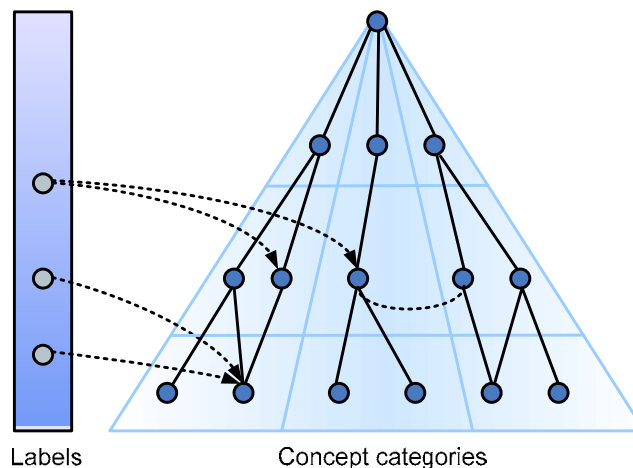


Figure 8. Mapping of human-readable symbols (labels) to the concepts in the ontology.

In this report, keywords are thought to represent overlapping and inexactly defined sets of entities in the real world. As such, each keyword refers to a fuzzy entity type, or in other words, to a fuzzy set. However, it would be too cumbersome or even impossible to define their membership functions, i.e. to evaluate the membership degrees of all entities in all sets in the domain. Neither would it be reasonable to ask the expert user to give degree values when annotating event reports. Instead, our approach is to use fuzzy versions of the is-a, part-of and dependency relationships between keywords.

In practice, experts might like to have some freedom in the use of keywords. Very often, words commonly used by industry experts have several meanings. For example, depending on the context the term "formation" could be interpreted as:

- a process phase performed by the web forming section
- physical and chemical phenomena occurring in the web forming section
- a quality measure (process variable) of the paper web.

In our approach, the keywords seen by the user have a one-to-one correspondence to the concepts in the ontology. We have not used identical keywords in different categories. As a consequence, the user should understand the fundamental differences between various keyword categories. To achieve this, the user interface of the KNOWMOBILE tool should support this understanding, for example, with a consistent appearance of the keywords.

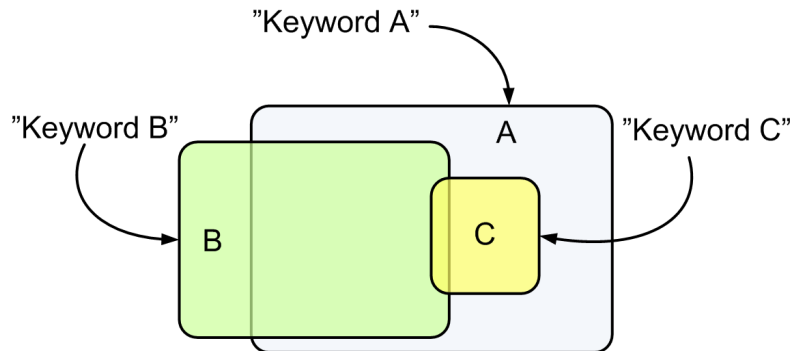


Figure 9. Inclusion and coverage among sets represented by keywords.

A common way to build fuzzy relationships is to associate a membership degree with each relationship instance. However, the "closeness" of two keywords in a taxonomy or in a partonomy may not be the same in both directions. The semantic distance between two keywords depends which one was given as query term. Furthermore, the taxonomies and partonomies must perhaps be processed in different ways. This is why we have adopted the approach illustrated in Figure 9. Here, set C is fully included in A. So, C can be understood as a subclass (subset) of A and all elements in C are a perfect match if A is given as a query term. In the other direction however, C covers only a minor part of A. Therefore, all elements of A may not be good matches if the user enters a query with the keyword C. With the help of this intuitive description we arrive at the following definitions of inclusion and coverage which will be used to define the fuzzy taxonomies and partonomies in the five keyword categories:

- Inclusion of B in A, $Inc(B,A)$: The degree to which the population represented by B is *included* in the population represented by A.
- Coverage of A by B, $Cov(B,A)$: The degree to which the members of B *cover* the population represented by A.

Considering keyword A to be more general than B and C we can draw Figure 10 which clarifies the inclusion and coverage usage in classifying keywords. The values of inclusion and coverage of two keywords can be expressed as real numbers between 0 and 1 or in qualitative terms, such as "minor", "moderate", "significant" and "total". In practice, these qualitative linguistic labels are assigned numerical values between 0 and 1, or more generally fuzzy numbers.

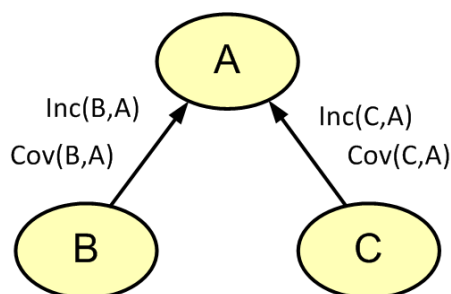


Figure 10. Keywords classified into a hierarchical network on the basis of their inclusion and coverage values.

In addition to the taxonomic is-a relationships, the concepts of inclusion and coverage can be applied to paronomies, i.e. to part-of relationships. For example, from the viewpoint of paper chemistry, "wet end" covers the major part of a paper machine and is fully included in it. The interpretation is, however, different from classification where keywords can be associated with sets of real-world entities. In the case of a part-of relationship between two keywords, inclusion and coverage express the overlapping of real-world individuals in the average. That is, "wet end is usually the most important part of a paper machine".

4.3 Presenting the keyword ontology

In order to develop a software tool the ontology must be expressed and stored in a more formal way. The basic approach for representing a fuzzy ontology is illustrated in Figure 11 with a combination of UML class and instance diagrams. From all kinds of nuggets, the demonstration only looks at event reports that usually describe problematic situations. Each report, like the specific instance "Report #1" in the diagram, is related to types of systems, plant functions, events, etc. Only the reference Report #1 to the

keyword "Holes" is shown in Figure 11. As indicated by the "instance of" associations, all keywords seen by the users are individuals, i.e. instances of a subclass of "Keyword category".

Fuzzy dependencies between keyword instances are described by a few fundamental relationship types like is-a (specialisation), part-of (here termed "partonomy") and, as an option, instantiation. In the first version we focus on "specialisations", i.e. fuzzy classification of keywords. The degree of overlapping (or inclusion) of the sets represented by the keywords is described by linguistic labels, i.e. natural language words like "moderate" or "significant". So, the instance named "Specialisation #1" in Figure 11 tells us that "Holes" is "to a large extent" understood as a subclass of "Quality problem but only represents a minor part of its scope". In addition, the keyword "Holes" may also specialise other problem types not shown in the diagram.

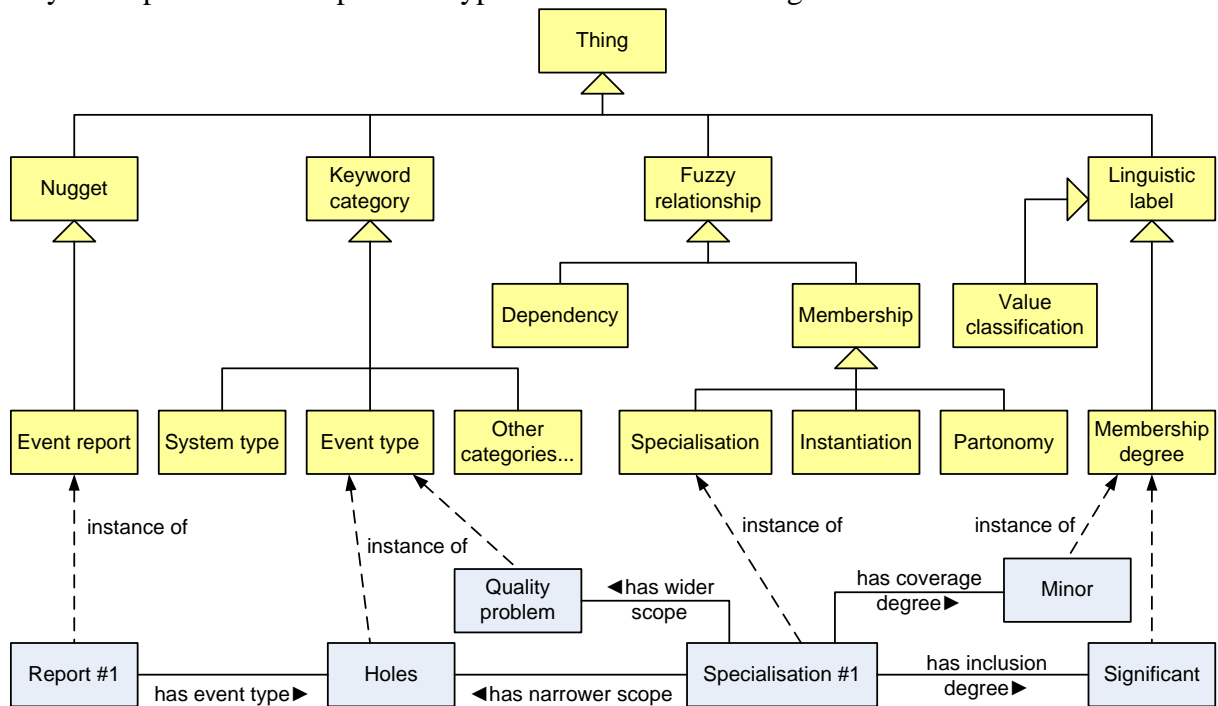


Figure 11. The idea of representing fuzzy keyword ontology with object classes and their instances.

To avoid unnecessary complexity, we modelled all kinds of dependencies with one single *dependency* relationship. For simplicity it is symmetric and has a strength value between 0 and 1. For example, the link between "wire section" and "formation" can be judged to be "significant". In addition to functional dependencies, the dependency relationship can be used to describe other types of links between keywords. For example, it can represent synonyms and dependencies created by systems located close to each other.

As discussed above, the two values of inclusion and coverage need to be defined separately for each pair of keywords. This leads to a large number of numbers which makes the development and maintenance of the ontology difficult. So, more user-friendly presentations are needed. Figure 12 shows an example of an "event classification matrix" that can make things easier for the ontology developer. Actually, the matrix represents both the inclusion and the coverage aspects of is-a and part-of relationships. Other categories, e.g. function types, have their own matrices.

The matrix shows keywords in column and row titles in the same order, broader terms first and terms considered being at the same level in a suitable order. This leads to a diagonal structure where inclusion can be shown in cells above the diagonal (yellow) and coverage in the lower part (blue). For example, all human errors are problems but they represent only 30% of all possible problems. Terms at the same level of a taxonomy or partonomy tree form blocks of cells around the diagonal.

Event type inclusion and coverage	Unspecified event	Problem	Technical problem	Operational problem	Quality problem	Human error	External event	System fault	Function failure	Web breaks	Holes	Omission	Fire	Device fault	Design flaw	Drift	Oscillation
Unspecified event																	
Problem			1	1	1	1	1										
Technical problem		0,8						1	0,5								
Operational problem		0,5			0,4				0,5	0,6							
Quality problem		0,5		0,5						0,3	1						
Human error		0,3										1					
External event		0,2											1				
System_fault														1	0,6		
Function_failure															0,4	1	1
Web breaks				0,5	0,3												
Holes					0,4												
Omission						0,3											
Fire							0,1										
Device fault								0,7									
Design flaw								0,4									
Drift									0,3								
Oscillation									0,2								

Figure 12. A simplified event classification matrix.

For almost all keyword categories there are both classification and partonomy matrices. The partonomy matrix is constructed in the same way as the classification matrix above. So, we have for each keyword category two networks, partially with the same nodes. The search for neighbouring keywords can jump at any node to the other network. This complicates the search. Furthermore, the dependencies between keyword categories allow the search to continue in some other category.

The sections below informally describe the keywords, their meaning and usage in the KNOWMOBILE tool, as well as some relationships between the terms. To define a sound "system of keywords", the goal was to first identify the most relevant entities used in searching event reports. In particular, types of problematic events and plant functions affected by them were considered interesting. As a next step, classifications of keywords were defined separately for each keyword category. For example, we tried to identify event types that are independent of process variables thus avoiding combined terms like "pH problem". The purpose of this orthogonality is to limit the total number of keywords. Instead, we characterised events by a combination of several keywords, such as event type, variables and functions affected, etc.

4.4 Principles of fuzzy queries

Basically, our demonstration is about intelligent searches for information in a knowledge base. The task can be illustrated with the following example:

- In the simplest case, the user gives only one search term, e.g. "event type \approx holes".
- As shown in Figure 13, the search should go along is-a, part-of and dependency relationships and try to find keywords that are close enough to "holes". This results in, for example, a list of related keywords with the best matches first, e.g. {"Pinholes", "Quality_problem"}.
- Terms in other keyword categories, e.g. "function \approx forming", can be combined by an AND operation. If oversimplified, this transforms into a "standard" query like:

((function = forming) OR (function = drainage)) AND
((event type = holes) OR (event type = quality problem))

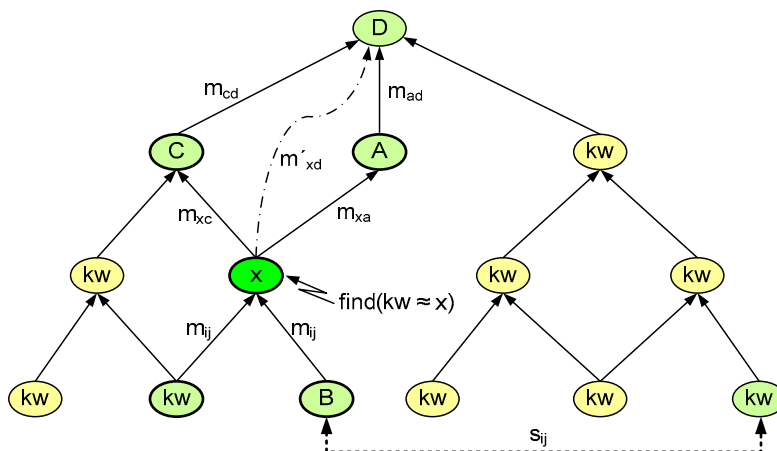


Figure 13. The search problem – finding related keywords in the ontology.

A simple procedure for searching the nugget database can be outlined as follows:

1. Define a query combining several keyword categories, for example:
(system ≈ "wire section") AND (function ≈ "drainage") AND (event ≈ "quality problem").
2. Find (separately) the closest neighbours and their matching degrees in each keyword category given in the query (system, function and event in this case).
3. Find all the nuggets that have the keywords either given directly in the query or mentioned in the corresponding list of closest neighbours.
4. Calculate a "combined matching degree" for each nugget found. Several approaches can be used to combine the matching degrees in each keyword category, e.g. the weighted sum of the system, function and event types.
5. Sort the nuggets according to their combined matching degrees.

So, the most important question is how to use fuzzy reasoning to find sufficiently relevant keywords. When this is done, the rest, i.e. finding the report instances and their actual content, could be implemented with more traditional methods. To solve the problem, the search algorithm should have, among others, the following features:

- When an expert attaches a generic keyword, say "quality problem", to a nugget, she/he can give it two interpretations: 1) this applies equally to all possible subtypes of quality problems; or 2) the expert doesn't know and selects, therefore, a more generic alternative. The first alternative might be suitable for guidelines and recommended practices, while the second one could occur in event reports that have not (yet) been analysed in detail. We are now looking primarily at problem reports, so option 2 can be taken as a starting point.
- The discussion above may affect the search algorithm. In case 1, the "closeness" of a specific search term X (e.g. "holes") to a broader term A depends on the inclusion of X in A. In case 2, however, it is unclear if A applies to X. Instead of A, the correct keyword of the report might have been a subclass of A other than X. So, the coverage of X in A should also be considered. If X represents only a minor portion of set A, the potential of finding good hits in nuggets with the keyword A becomes smaller. We might say that the closeness of A to the given search term X is equal to the coverage of X in A.
- If a generic keyword X is given for a query, how can we determine the closeness of a more specific term B? We can assume that the expert wants to find all events belonging to any subtype of X. Therefore, the closeness of B to X is directly the inclusion of B in X.

- The search should process upwards and downwards in the tree. Obviously, the closeness of more distant nodes can be obtained by combining the closeness values of each neighbouring pair of keywords, for example by multiplication or a fuzzy AND operation. However, due to the multiple inheritance there may be several paths between nodes. For example in Figure 13, the top node D can be reached from X through A and C. So, D obtains two closeness values that should be combined by using operators like maximum (fuzzy OR), minimum, average, etc.
- What was said above concerning taxonomies obviously applies to part-of relationships and instantiations, as well. However, we can assume that when the expert gives the keyword "paper machine", he/she thinks of an event that concerns the whole machine or a major part of it. Small parts like individual process components are not relevant. Therefore, the "closeness" of a whole and its parts can be approximated by multiplying the inclusion and coverage values.

Other, horizontal dependency relationships within and between keyword categories are not considered to be *is_a* or part-of relationships. Instead, they can represent, for example, functional or geographical dependencies or some other kinds of "similarities" between keywords. These relationships can be modelled by symmetric dependency instances with only one degree value for both directions.

4.5 Event types

In this document, the term *event* refers to an interesting period ("episode") in the operation of an industrial plant. Therefore, an event has a duration that is usually rather short but can continue for weeks or even months. Most often, an event is interesting (i.e. valuable for knowledge management) because it is unanticipated and unwanted, i.e. a problematic situation. However, also exceptionally good operating periods or test runs can be reported as events. It is also foreseen that any practical information system should be able to describe complex event chains, e.g. the causes and consequences of the "primary" event (Figure 14). In other words, part-of relationships could be applied to events, as well. The KNOWMOBILE demonstration tool, however, looks only at one event at a time (the primary event C) omitting its causes and consequences.

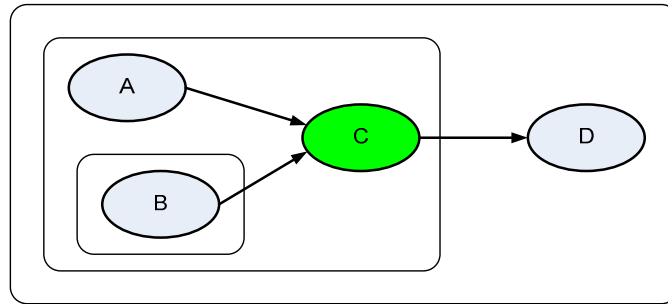


Figure 14. Complex events consist of lower-level events.

Many event types characterise the dynamics or behaviour of the state of a system (more generally the "state-of-affairs") as compared to the target state. For example, a process variable may be too high or vary in an uncontrolled fashion. For classifying events we first selected a few relevant characteristics that were used for dividing the population of events into categories, for example:

- Value of the event: is the episode desired or unwanted? Its value to the plant owner can be positive, negative or neutral.
- The kind of plant component associated with the event: technical system, plant function, substance, people or external entity.
- The character of the operational state that operators try to maintain or achieve: stationary or transient.
- The kind of action triggering the event: intentional, unexpected.

These questions were used for guiding an expert evaluation that leads to a classification of event types. Figure 15 shows a fragment of such a classification of events. At the top level, a generic event is classified into problems, neutral observations and successes on the basis of the event value. At the next level, associated plant items are used to categorise problems into more concrete event types. The inclusion and coverage values introduced in section 4.2 are not shown in the figure.

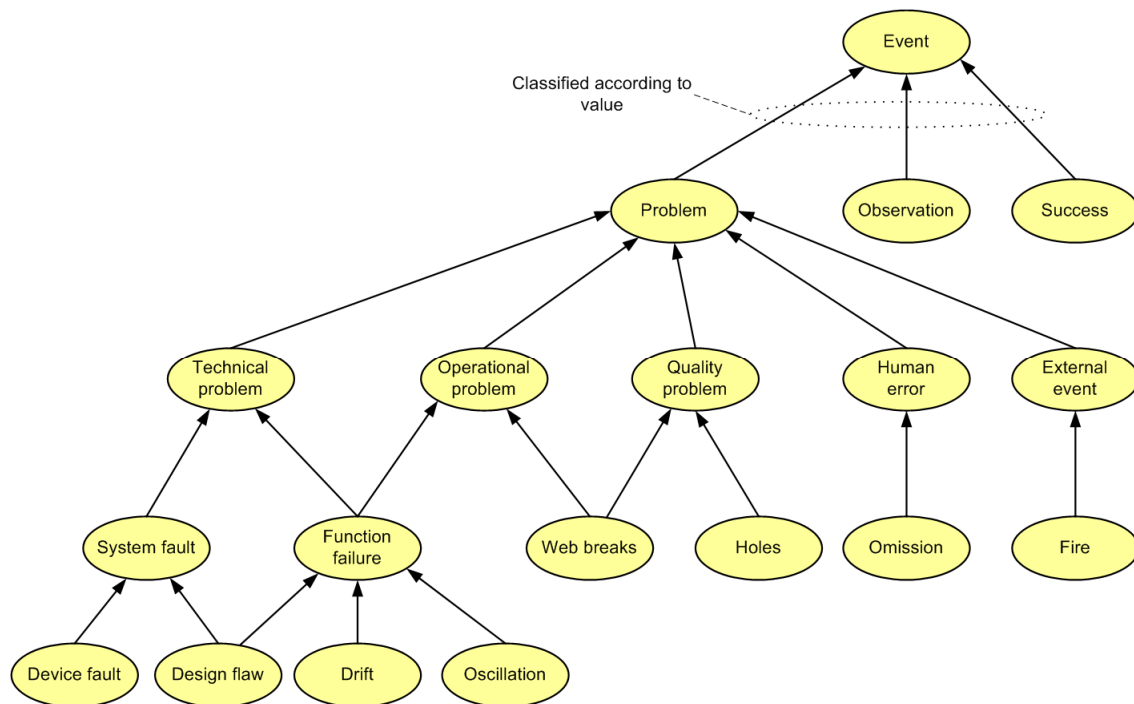


Figure 15. A fragment of the event type classification.

4.6 Systems at an industrial plant

A system is considered here to be a real-world (physical) entity that is designed for a purpose. Systems consist, for example, of mechanical and electrical equipment, software and people. Systems can be seen as resources that can be used to carry out a task, e.g. a production process or a monitoring and control task.

Figure 16 shows a few examples of generic system types. In our application, technical systems are in the focus, but organisational units like companies and teams should also be an option. Complex socio-technical entities such as paper mills consisting of both technical systems and organisational units are needed to indicate the overall context of an event.

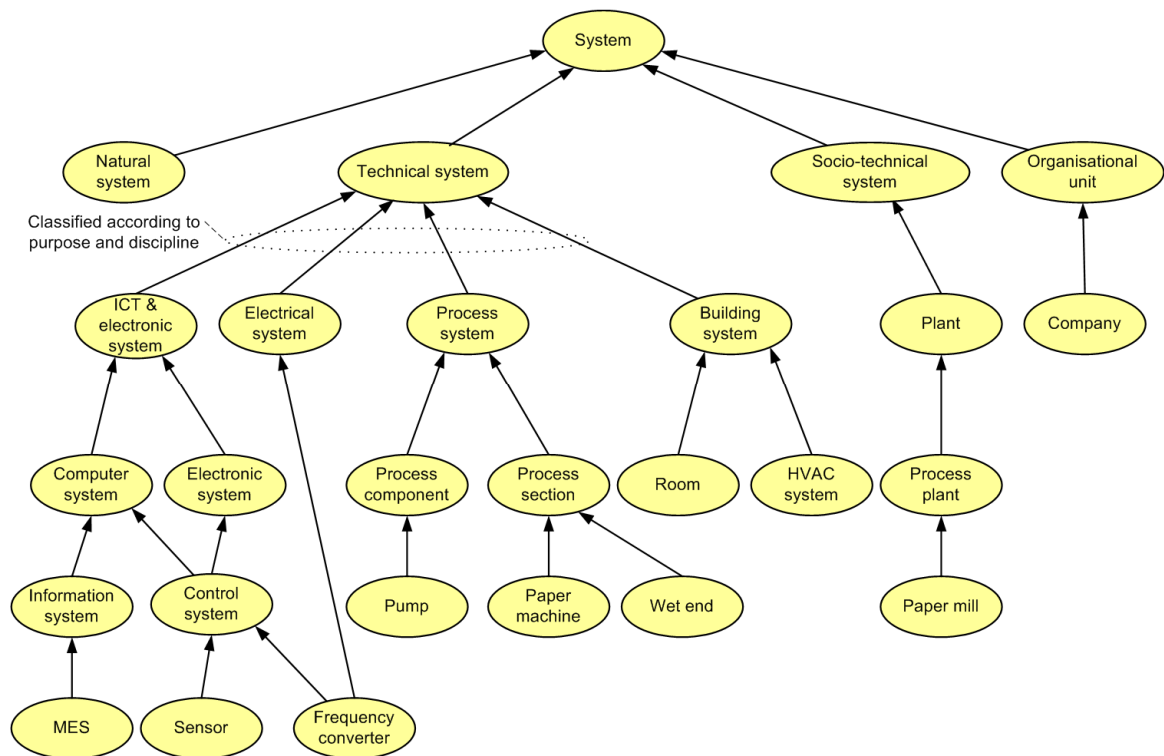


Figure 16. Classification of system types, examples.

A possible dimension for categorising technical systems comes from their primary purpose and engineering discipline. The role of *process systems* is to transfer and transform process materials. In other words, process systems are more or less identical to the term "process equipment". They belong to the realm or process and plant engineers. Electrical systems supply electrical energy, and building systems provide a favourable operating environment for the systems and people at an industrial plant. The main purpose of "ICT & electronic systems" is to monitor and control the manufacturing processes and systems of a plant.

Most of the generic terms shown in Figure 16 are not normally used as keywords. However, this should be possible if, for example, the specific term is not known at the time of writing. Consequently, top-level keywords up to the word "system" should also be available to the expert user.

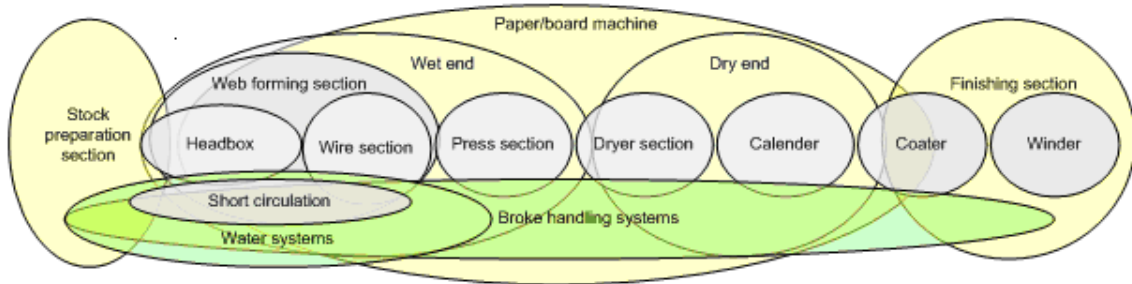


Figure 17. An example – fuzzy decomposition of a paper making line.

As indicated above, a *system* is an assembly of real-world (mostly physical) objects serving a common purpose. A system usually consists of subsystems and finally of atomic components at the lowest level of decomposition. The boundaries of a system depend on the viewpoint taken, and even components can sometimes be considered as systems of their own. Therefore, we are not making a distinction between systems and components in our keyword classification. Both can be understood as "system elements". Figure 17 shows some subsystems of a paper making line and also demonstrates how system decomposition often has a fuzzy character depending on the viewpoint taken. Maintaining the correct viewpoint may be difficult at some points. Frequency converters, for instance, can be considered as actuators, and, therefore, as "control devices". Quite obviously they also belong to the domains of industrial electronics and electricity distribution and could be called "electrical equipment". In addition to that, frequency converters are parts of both control systems and electricity distribution networks.

Consequently, both *is-a* and *part-of* relationships are important for systems. They must, however, be kept apart from each other. Figure 16 above illustrates some specialisation paths. Partonomies could have similar but separate representations. The semantics of specialisation and *part-of* relationships are different, which can have implications for the inference rules of a "fuzzy keyword engine". This was discussed in more detail in section 4.4.

4.7 Plant functions and activities

An industrial plant is a complex system that is supposed to perform a production process. To accomplish the overall production task, many kinds of activities are required, for example:

- managerial tasks
- processing of raw materials and other substances
- control of plant equipment, e.g. starting machines
- control of process variables
- maintenance activities.

Here, these activities are called plant *functions*. In many cases, a function refers to a purposeful activity of a goal-oriented agent like a human operator, a control system or process equipment. Occasionally, functions can be understood as physical and chemical phenomena that are being controlled.

As mentioned above, an industrial plant performs various tasks to accomplish the overall production. Figure 18 shows some examples and their classification into "Operation and Management", "Processing", "Control" and "Phenomenon". There is a difference between phenomena and intentional function types like "Processing". Roughly speaking, phenomena, for example flocculation and coagulation, take place in the process equipment according to the laws of nature. Many phenomena are affected by plant operations and also controlled by them. However, here we do not differentiate between "Processing" and "Phenomenon".

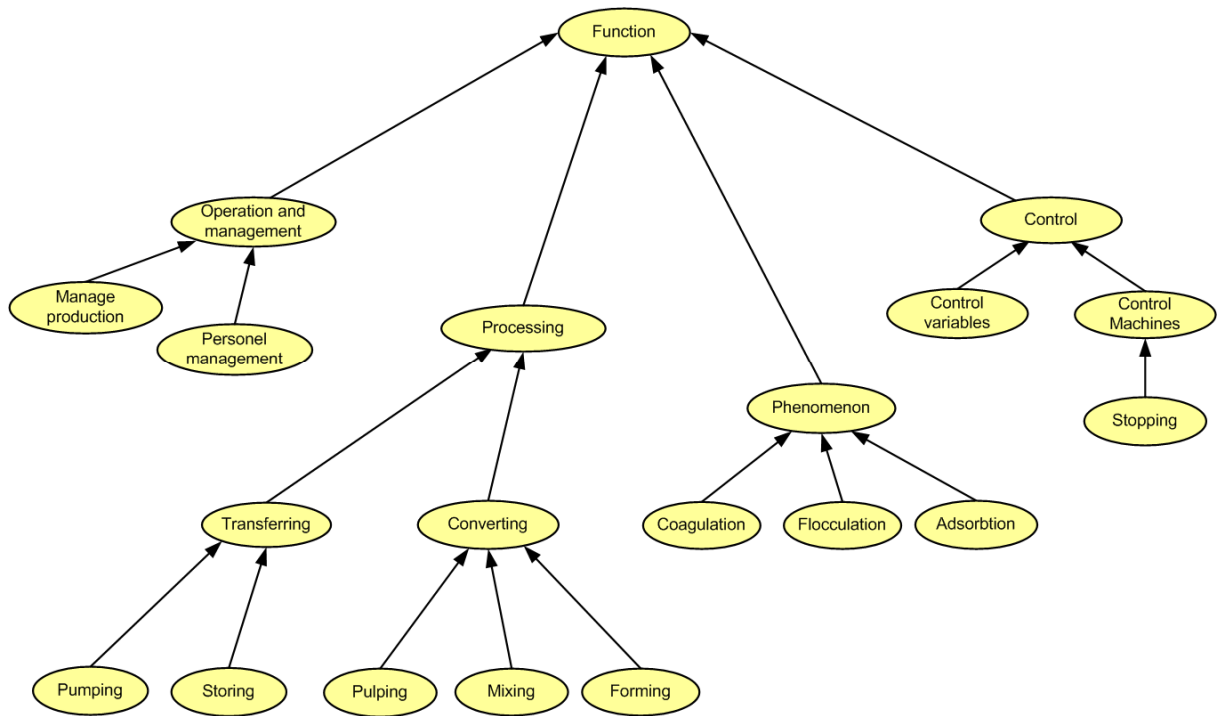


Figure 18. Examples of function type keywords

Like physical systems, functions can be decomposed into smaller parts. For example, the process of "Paper making" consists of "Pulping", "Stock preparation", "Web forming", etc. This adds one or more (fragmentary) partonomy trees to the keyword ontology.

4.8 Process variables

The term *process variable* refers to various attributes of plant systems, functions and substances that characterise their performance or state. They are not necessarily measured or calculated by a control system but can be obtained, for example, from samples or visual observations. Therefore, variables can have a very qualitative character without a numerical scale.

Variables can be classified as illustrated in Figure 19. Even limited partonomies can be easily identified. For example, the variable "Retention" describes the portion of fibre and filler components ending up in the paper web. "Retention" can be viewed as a sum of various parts, "Ash retention" being one of them.

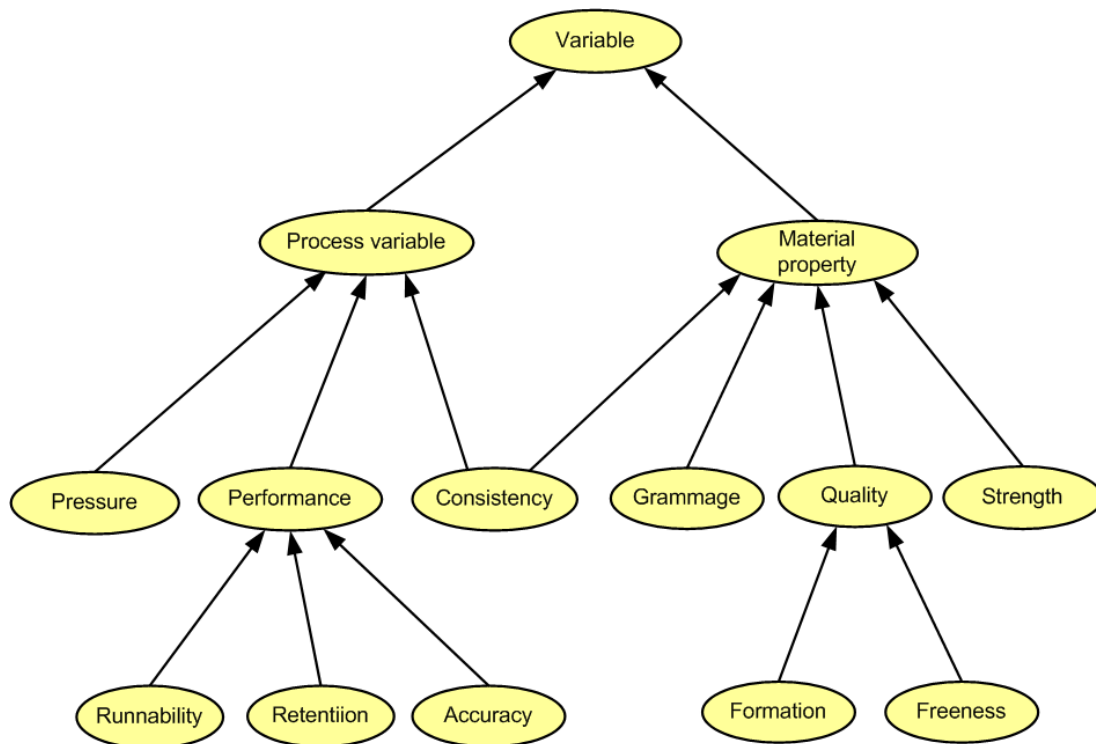


Figure 19. Examples of variable types.

Variables can be used in two ways. Firstly, variable names can be added as keywords in order to say that an event is in some way associated with the variable. Secondly, variables and their values can characterise the situation during (the course of) an event. With the values we have, again, two options. Exact numerical values would not support reasoning because the KNOWMOBILE tool doesn't know which values should be considered low and high in a given process and operational state. The second alternative is to allow the expert user to decide upon and assign a *linguistic value classification label*

like "normal", "high" or "very low" to a process variable. In the current version of the KNOWMOBILE tool this feature has, however, not been implemented.

4.9 Materials and substances

An industrial plant processes and handles many kinds of substances that have various chemical and physical properties and purposes in the production chain. Some of them are usually involved in an event.

Figure 20 shows some examples from the fuzzy keyword classification of materials found in industrial plants. Several criteria can be used for their classification. Here, the taxonomy is based on two dimensions, firstly the purpose that a material has in the production process, and secondly, it's physical and chemical properties. The purpose of a material is, of course, highly dependent on the application domain. The generic keywords "Process material" and "Substance" represent these two main branches of the tree. Below them, one can find commonly used names of products and chemicals. At the lowest level of the taxonomy, there are names of manufacturer-specific products. Also fragments of paronomies can be identified. For instance, the raw material of a paper machine is a mixture of several components like "Process water", "Pulp" and "Additive".

4. A fuzzy ontology for process industry

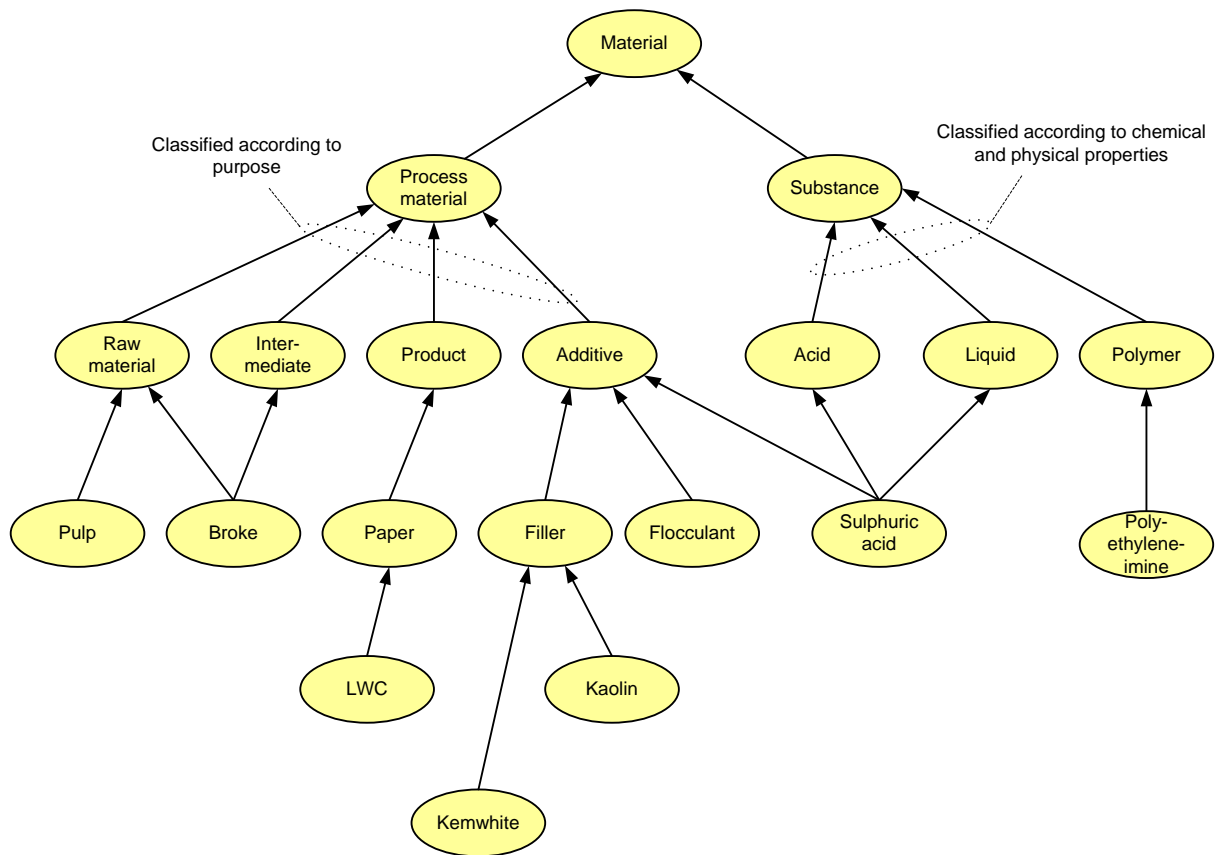


Figure 20. Examples of material types.

5. A demonstration application

In order to evaluate the feasibility of the proposed fuzzy ontology in an industrial setting, a demonstration application was constructed. The setting for the demonstration was that of an industry expert trying to solve a problem related to the process control and chemistry of the wet end of a paper mill. In solving such problems, the expert would benefit from knowledge of similar problems (and their solutions) in the past. The demo application provides efficient retrieval of past problem-solving situation reports by extending the information query using the fuzzy ontology. By extending the query, the expert can retrieve and benefit from past reports, even if the context of the past situation is not exactly the same (for example, slightly different process equipment, variable or chemical). Actual problem solving reports were collected and annotated for the demo with the help of industry experts.

5.1 Demo architecture and implementation

The demo was implemented with Java, using the Protégé ontology editor to maintain the fuzzy ontology in OWL format. For practical reasons, the demonstration was implemented as a standalone application. In a real-world setup, one would expect the core of the tool to be embedded in a web server, for example by using techniques like Service-Oriented Architecture (SOA) and Enterprise JavaBeans (EJB). Nevertheless, the demo was based on a component-based architecture (as shown in Figure 21), allowing us to easily compare and experiment with different reasoning approaches.

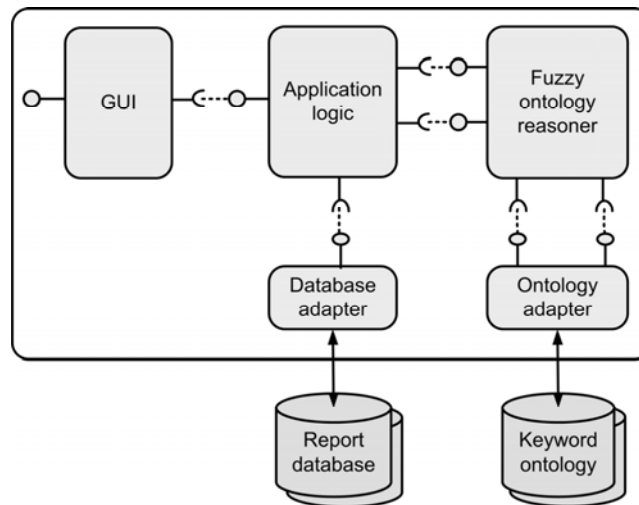


Figure 21. The component-based demo application architecture.

The **GUI** component (Graphical User Interface) guides the user in specifying the information query, and presents the results. Tools for browsing and evaluating the fuzzy reasoner component directly were also provided.

A **database adapter** is used to access report data, which in this case was stored locally in XML files. Similarly, an **ontology adapter** is used to provide access to the fuzzy ontology, in this case stored in OWL files. The adapters help hide the different interfaces and protocols of different data sources (e.g. SQL, HTTP) and provide transparent access via an agreed interface.

The **fuzzy ontology reasoner** component is used to process ontology-based information. Its main function in the demo is to extend a list of query keywords to a list of their closest neighbours in terms of fuzzy ontology relationships. For maintenance and evaluation purposes, the component interface also provides methods for directly accessing the ontology concepts and relationships.

Finally, the **application logic** component binds all the functionality together by taking the query, using the reasoner component to extend it, passing the extended query to the report database and then combining and ordering the results for the GUI.

The fuzzy ontology with fuzzy concepts, relations, and instances was defined using Protégé version 3.4. The developed ontology was exported for the reasoning software as a standard OWL file. Protégé is a widely used tool for developing ontologies. As such it provides some advantages, for example a forms-based interface for editing the basic classes and adding the individual keywords, reports, as well as their relationships (Figure 22).

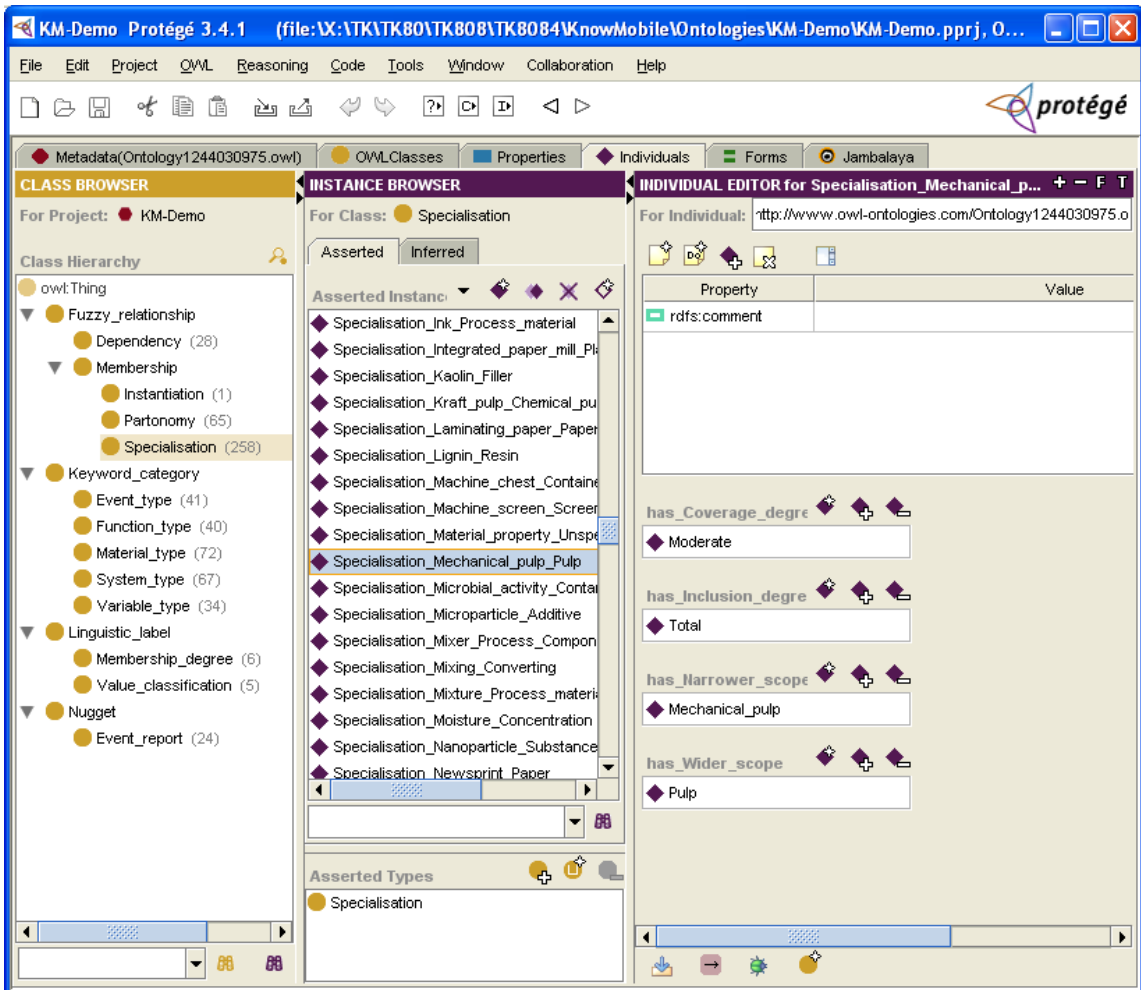


Figure 22. Adding a fuzzy specialisation relationship between material types "Mechanical pulp" and "Pulp".

The problem is that Protégé gives no built-in support for modelling uncertainties. All fuzzy concepts must be added on top of it, for example in the way described above. As a consequence, the number of keywords and relationship instances tends to become rather large. For each keyword, several specialisation, partonomy and general relationships may be needed. Managing modifications by hand and keeping the knowledge base consistent will soon become impractical for a real-world application. So, at least part of these tasks should be automated.

5.2 Demo user interface

The demonstration was provided with a graphical user interface to access tools for defining the query and browsing the results, but also for exploring and debugging the fuzzy ontology. The auxiliary "maintenance" tools allowed us to see the fuzzy process-

ing in detail, and discover inaccurate fuzzy relationship definitions (that can result in illogical keyword matches in the query extension phase). Figure 23 shows an overview of the user interface with several tool windows open.

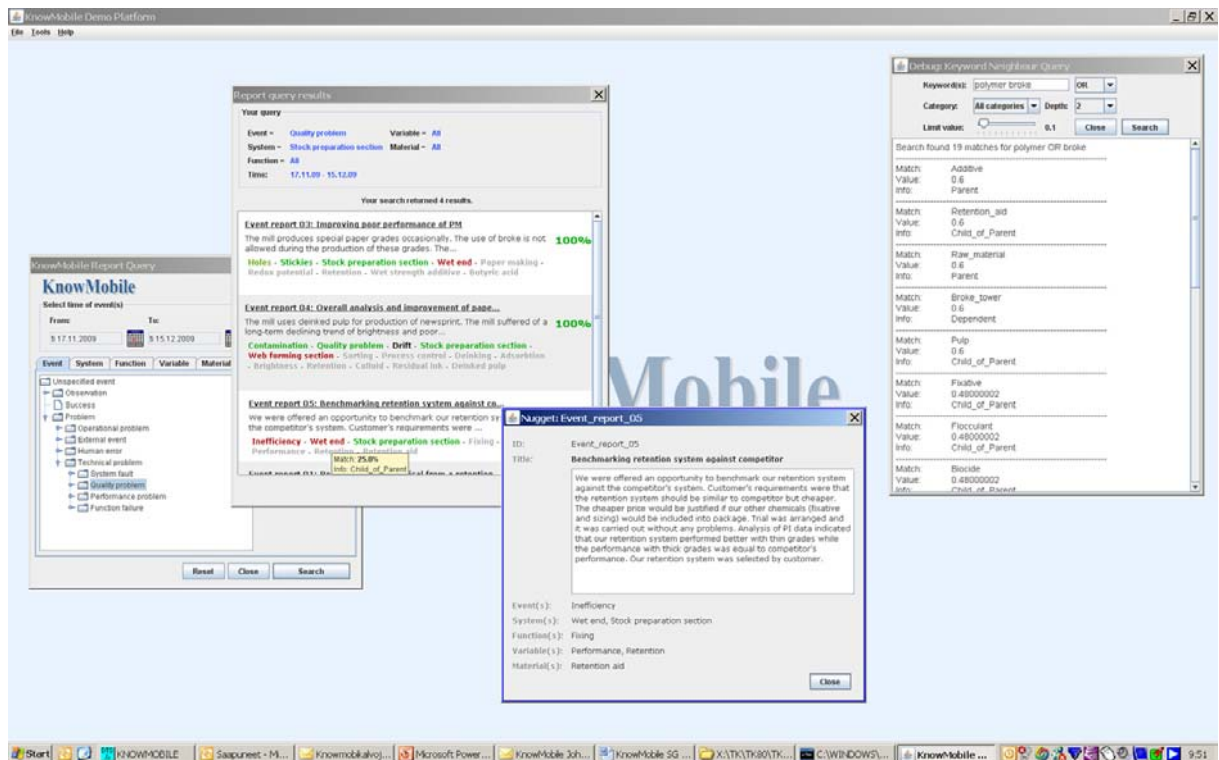


Figure 23. The demonstration application provides tools for searching event reports and (for maintenance purposes) exploring the fuzzy ontology.

The dialog window for defining a fuzzy query is shown in Figure 24. To facilitate an ontology-based query, the user has to select suitable keywords from a predefined set. Accordingly, the user interface should help the user to quickly find the appropriate terms. The tool assists the user in several ways. The keywords are arranged in taxonomical trees, so that the right keyword can be found by moving from generic, upper-level concepts to more specific subclasses. The user can also locate known concepts quickly with a free text search. To further assist the user, mouse tooltips display a description of each keyword.

The query keywords are selected from categories such as event (e.g. "instability"), system (e.g. "paper machine" or "head box"), function (e.g. "water removal" or "hydration"), variable (e.g. "pH") and material (e.g. "sodium hydroxide" or "packaging paper"). For each category, several (or no) keywords can be selected.

First, the user can select the period of time from which similar situation reports are of interest. The user may wish to, for example, 1) discard reports too old due to changes in the physical process setup that render the old reports irrelevant or 2) focus on a certain

time period during which memorable situations of particular interest to the user had occurred. Shortcuts for typical time frames such as "past month" or "all" (for the entire history) should also be provided.

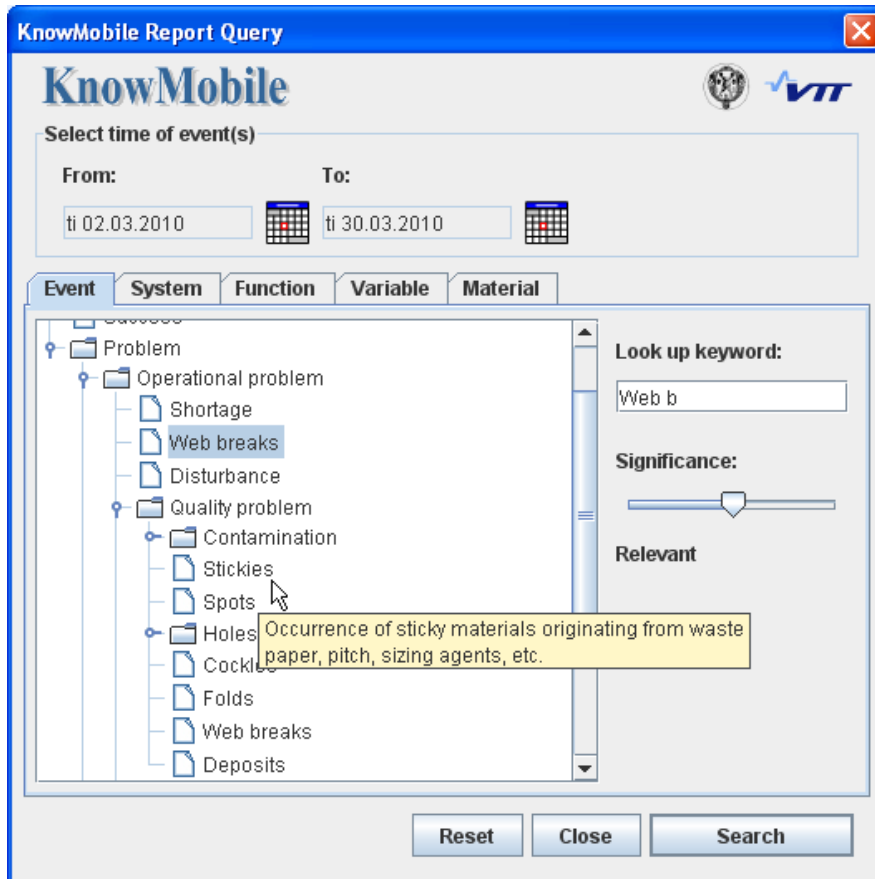


Figure 24. The application window for specifying the query.

Next, Figure 25 shows the window for displaying the results of the fuzzy extended search. The results are listed in the order of how "good" the matches are. The degree of relevance of each result should be expressed in a form that is universally understood (e.g. a value from zero to hundred instead of fuzzy constructs). Colours are also used to illustrate how well both the results and the individual keywords in each result nugget match the keywords selected in the query.

The first result in the exemplar output in Figure 25 has 100 % match value, since for each category the report has been annotated with at least one of the exact query keywords. The lower results, however, have only approximate matches in some category. As the match value of a keyword decreases, its colour shifts from green to red. As the particularly poor aspects of each result are thus highlighted, the user can now quickly determine whether the result is of interest. If there is a difference in the keywords that is

critical (e.g. the chemical involved *must* be calcium sulphate), it is quickly noticed and the result ignored.

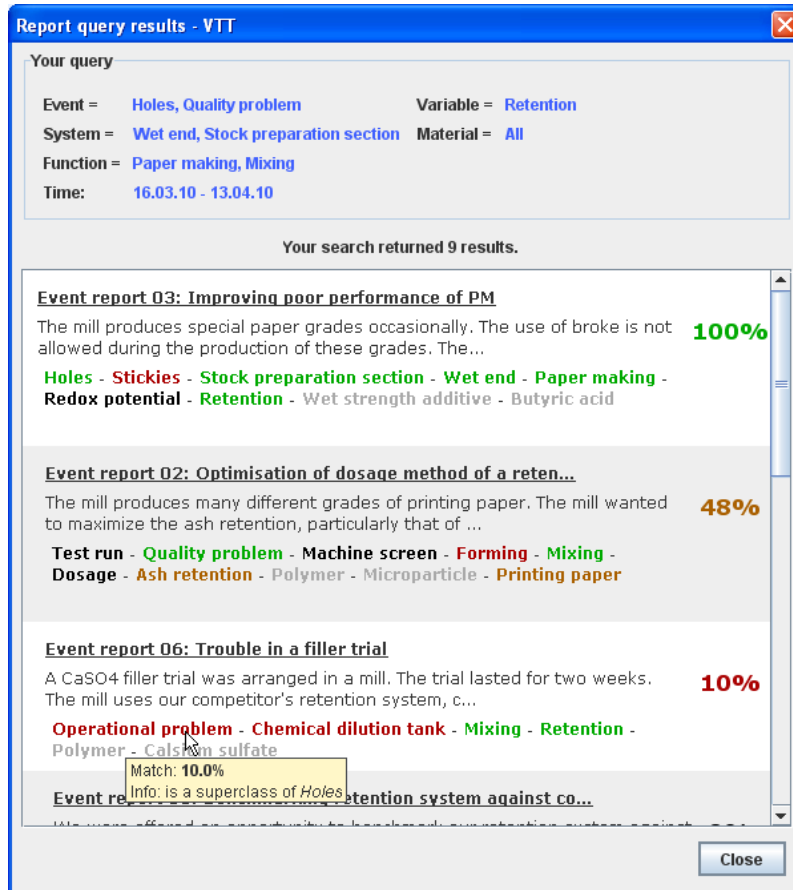


Figure 25. The demo application displaying the results of a fuzzy extended search.

To further explain the reasoning behind the result matching, the user is also provided with some insight into the fuzzy reasoning process. Hovering the mouse over a result keyword reveals a tooltip which, in this case, explains that the individual result event keyword "Operational problem" has a 10% match to the query, because it is a superclass of an event mentioned in the query keyword ("Holes") with minor (0.1) coverage.

Finally, Figure 26 shows the tools for directly testing the fuzzy ontology, used both for maintaining and verifying the ontology relationships. A graphical tool can be used to view the relationships as trees. The keyword "Quality problem" appears as a subclass of both "Operational problem" and "Technical problem", and the keyword description is given in the mouse tooltip.

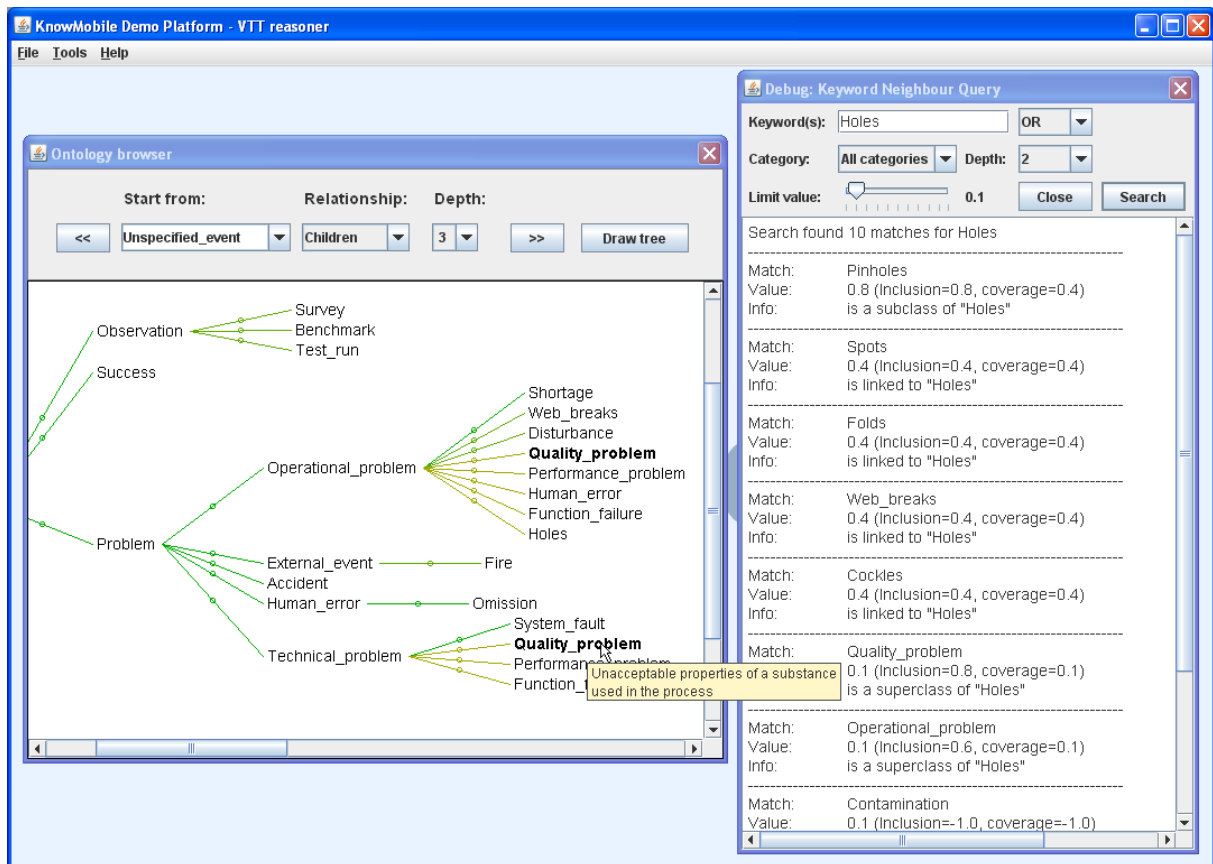


Figure 26. Demo tools for directly evaluating the fuzzy ontology.

Also depicted in Figure 26 is a tool for finding the fuzzy neighbours to any given keyword. The neighbours are listed in the order of relatedness, with the fuzzy relationships described in text. For example, "Quality problem" is listed as a related keyword for "Holes", being that it is the superclass concept. The associated weight for this relationship is 0.1, which corresponds to "minor". The user can also change some of the parameters of the fuzzy reasoning engine in order to see their effects on the results.

5.3 Extension to an external information source

The first version of the demonstration used 24 problem solving reports that were specifically collected and annotated by domain experts specifically for the purposes of the demonstration. After the first version was demonstrated, the decision was made to experiment with a connection to an external database of reports collected by KCL, a research laboratory that was formerly owned by the Finnish pulp, paper and board industries and now is part of VTT. This was motivated by several reasons:

- Instead of a small set of manually annotated documents, the KCL database contained about a thousand report abstracts that have already been annotated with suitable keywords.
- The KCL report keywords were selected from the Paperbase terminology which, although it is not a proper ontology, still contains some relationships between concepts. The ability to combine knowledge from several sources with different conceptual models is a key research issue for further demo concept development.
- Due to the recent merging of VTT and KCL, researchers at VTT had access to confidential KCL data.
- There already is a search tool called Ultraseek that can be used to query the KCL reports. For demo evaluation purposes, Ultraseek provided a point of reference, a way to compare the demo search results to a more ‘traditional’ way of retrieving similar information.

Figure 27 displays a screen shot from the Paperbase library (<http://www.paperbase.org/>) with some of the Paperbase terms listed. In all, there are about 7700 terms. Relationships such as "broader term" (superclass), "narrower term" (subclass), synonym, and generally "related term" have been defined for some of the terms.

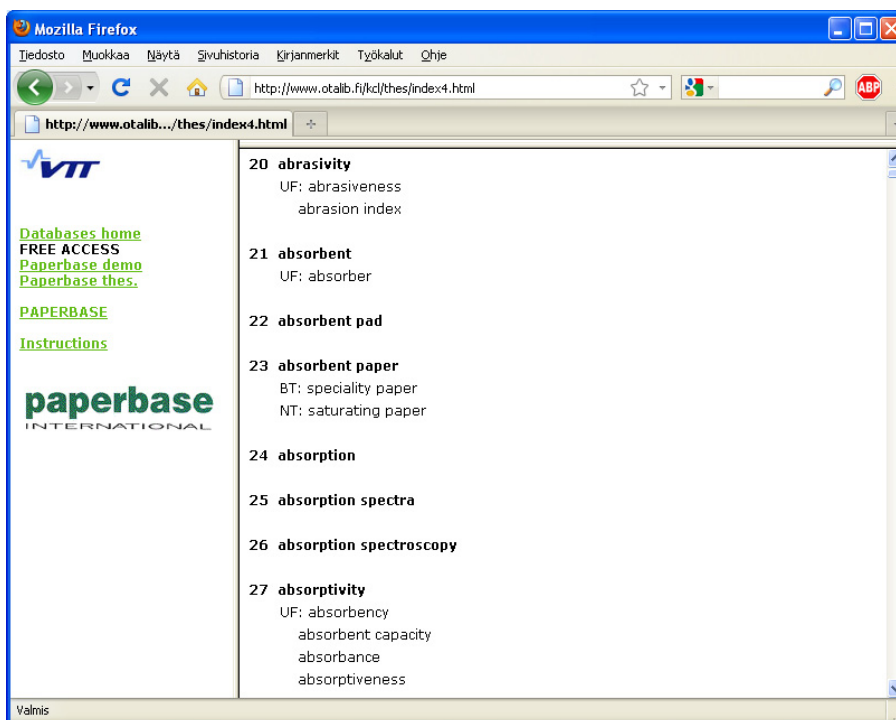


Figure 27. An excerpt from the Paperbase terminology.

The original demo architecture and the specified component interfaces did not address the issue of accessing several different data sources with different ontologies or conceptual models. As a result, implementation of the connectivity demo - in terms of both the structure and the ontology reasoning process - was rather rough and experimental. In the original setup, the component "database adapter" takes care of access to a data source. To support access to a different data source, the adapter was re-imagined as a sort of software agent, capable of "wrapping" the source – translating the queries to the concepts used in the source database and thus providing transparent access. The adapter interface was implemented with a new class KCLReportAccess, for which new methods for translating concepts and queries were defined.

In the connectivity demonstration, the extension of the nugget query proceeds as follows (see Figure 28):

- Extend the original query to neighbour terms specified by the fuzzy ontology (as before).
- Pass the extended query to the "remote" KCLReportAccess component.
- Within the "remote database", translate the query terms to the concepts used in the Paperbase ontology. For this, a simple cross referencing matrix was defined. Note that all keywords do not have suitable counterparts in the Paperbase terminology, while others have several.
- Furthermore, use the concept relationships defined in the Paperbase terminology to further extend the query to related keywords.
- Finally, query the KCL report database and report the results, explaining how and why the query was extended and translated in each phase.

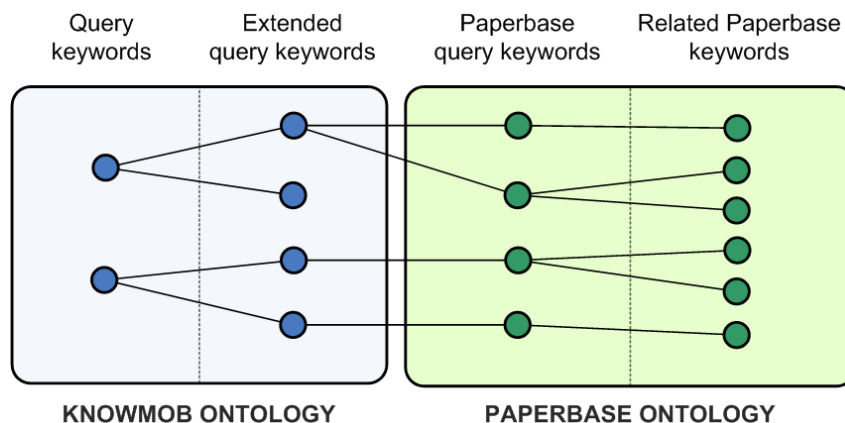


Figure 28. Mapping the original query to concepts used in a remote knowledge base.

To be able to handle the Paperbase ontology, the KCLReportAccess component was provided with a class PaperBaseOntology, itself an extension of the ontology manager class specified in the original demonstration.

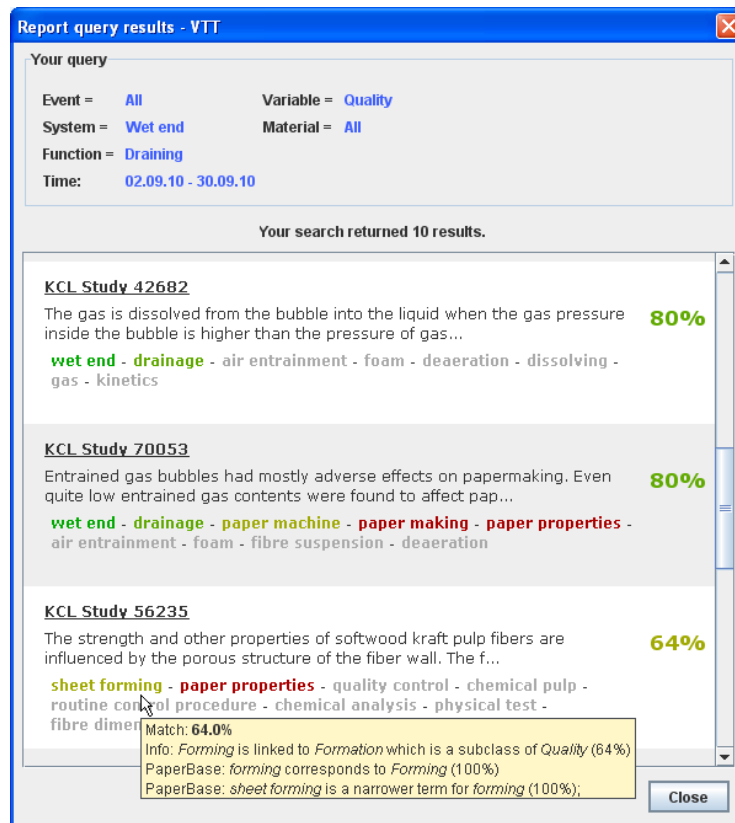


Figure 29. Search results from an external knowledge base.

For a screen shot of exemplar query results, see Figure 29. The mouse tooltip for the keyword "sheet forming" sheds some light on the query extension logic for two ontologies. First, the KNOWMOBILE ontology extends the query keyword "Quality" to the related keyword "Forming" via the subclass "Formation". Then, cross-referencing "Forming" results in the Paperbase term "forming", the only difference being the lower case first letter. Further, the Paperbase ontology contains a subclass relation from "forming" to "sheet forming". As a result, the Paperbase term "sheet forming" is deemed a fuzzy neighbour of the KNOWMOBILE term "Quality" (with, in this case, a match rate of 64%).

5.4 Evaluation of the demonstration

The demo was evaluated with industry knowledge management experts from VTT. The general feeling was that some of the exemplary fuzzy search results seemed useful.

However, there was significant doubt as to whether the cost of constructing and maintaining the fuzzy ontology is prohibitively high. Combining different ontologies is an interesting topic – although the KCL reports that were used as case material were products of researchers, operators of the plant could still benefit from them despite a different domain view. Other observations included:

- The "relatedness" of some of the concepts' neighbours was set unjustifiably high. Obviously the relationship weights would need some thought.
- The demo application should include a feature allowing the user to search for fuzzy neighbours of individual search results ("find more like this").
- A real application should let the user influence the processing. If users were able to react to a match that does not make sense (or a useful match that is deemed poor), user feedback could then be used to "teach" the fuzzy ontology.
- The query keywords in one category have an OR grouping, which in some cases can be unpractical for the user. Perhaps AND should at least be an option. In a way, the fuzziness already serves the same purpose as OR – extending the query.
- The order in which the query keywords are selected in the query or listed in the result nugget should perhaps have an effect on match values. Also, the overall match value calculation does not consider the number of keywords used in the result nugget. Perhaps if the set of keywords used is smaller, the match value should be greater, and vice versa.

Regarding the connection to the external (Paperbase) ontology:

- If we are to develop the demo further, surely the aspect of dealing with different ontologies is very relevant. Accordingly, the heterogeneity of the information sources and mapping of ontologies should be taken into account at the demo concept level and in the demo application architecture.
- Both the cross-referencing to the Paperbase ontology and the extension to related Paperbase concepts should obviously have an affect on the keyword match values and therefore the overall nugget match assessment. Currently, little effort has been put into defining the cross-referencing matrix, and the related Paperbase keywords are given a fixed match weight.
- Properly assigning the match weight for the Paperbase term relations will likely result in even sets of weights that should somehow be easily defined and updated.

The query keywords are translated to Paperbase terms, but the keywords in the results as they are shown are not translated back to the terms specified in the KNOWMOBILE

ontology. This was necessary since the Paperbase ontology contains about 7700 terms, most of which obviously are unknown to the KNOWMOBILE ontology. For better understanding of the query results, the keywords of the source database are displayed as such. It is unclear whether this is exactly how the application should work.

5.5 Ideas for further development

The demo concept presumes that all the reports in the knowledge base have been properly annotated. However, no tool for such purpose was implemented. Here, we present some ideas of what such a tool could look like, if we assume that the annotation is a task that can and should be done manually.

The worthwhile pieces of knowledge (or "nuggets") that are stored in the company knowledge base must be annotated with descriptive keywords to facilitate fuzzy search. The contents of the nuggets themselves can be of whatever format; text, trend graphs, video, etc. can be easily (drag-and-drop) incorporated by the user, since it is the annotated metadata that the fuzzy search is based on.

After the user has collected and attached or linked all the necessary information, suitable keywords are selected to describe the situation report. Figure 30 presents a mock-up interface for **nugget metadata annotation**. First, managerial and background information such as identification number, report author name, title, and time of the event are selected. A short description can be inserted. Then, the appropriate keywords are selected.

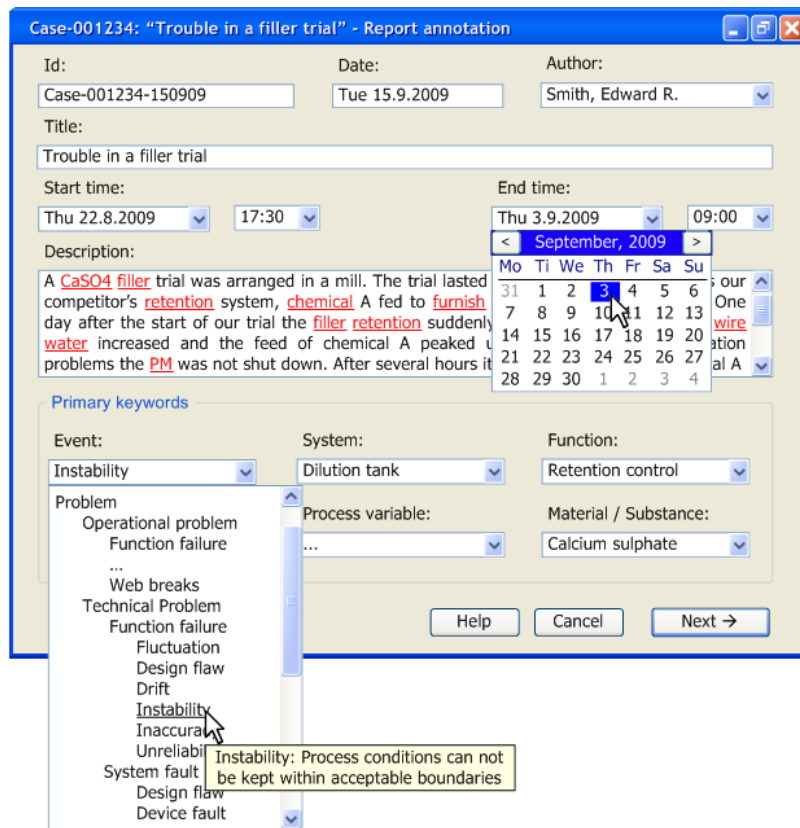


Figure 30. Concept of a user interface for nugget metadata annotation.

Here, the keywords are selected from a drop-down-menu. The keywords are again arranged according to their taxonomy, so that the right keyword can easily be located by moving from generic to more specific terms. You will note that the list is constructed based on a fuzzy ontology, since multiple inheritance is shown (e.g. "Design flaw" is shown as a subclass of both "System fault" and "Function failure", and "Function failure" itself as a subclass of both "Technical problem" and "Operational problem").

The number of necessary keywords in each category may vary. An empty value may also be selected to signify that a term from that category is not relevant in describing the nugget. Also, more than one keyword may be selected from each category (for example, several substances or process variables may be involved).

The system should – as far as practical and feasible – support the selection of keywords by analysing the information stored in the nugget. In Figure 30, for example, the system has identified certain terms (highlighted in red) in the report description, and uses them to automatically pre-select keywords for the system (dilution tank), function (retention control), and material (calcium sulphate). Also, it should not be possible to select keywords that do not make sense in the same context with each other.

The use of drop-down-menus for the selection of a suitable keyword is one option. In Figure 31 we present another option for keyword selection in the context of the **query**

definition. In the example, "Web breaks" is selected from a dynamic map that illustrates the classification of events. Because the number of available keywords can be staggering, the user must be assisted in finding the particular keyword(s), e.g. by advancing from more generic keywords to more exact subclasses. Since fuzzy ontologies enable multiple inheritance, the keyword "Web breaks" can be discovered through different taxonomy branches, once again making it easier to find. Furthermore, the user may wish to specify the significance of each category if some are more relevant than others.

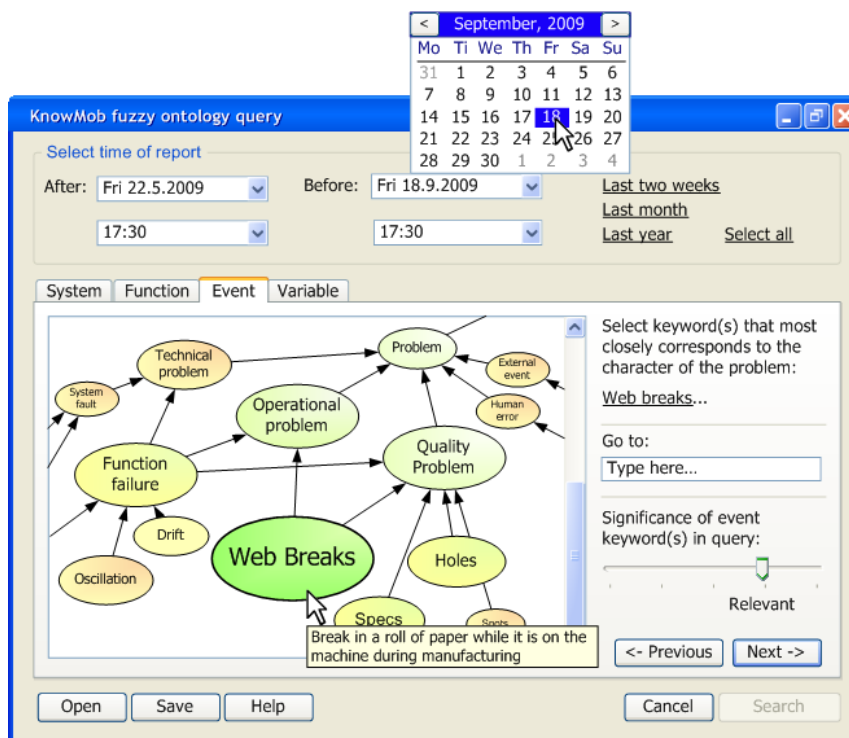


Figure 31. Concept of a user interface for selecting the query keywords.

It is worth noting that all the relationships between the concepts are not displayed in the 3D map. Other relationships such as paronomies or general relationships may be taken into account by for example using colours to highlight concepts that are related to each other. The objective here is to find the right keyword, and it is for this purpose that the taxonomical relationships are very useful.

The interface for **result browsing** should provide easy ways for the user to modify the original query if, for example, the results indicate that some of the keywords were poorly chosen.

Another idea for further development is to ask for feedback from the user. The knowledge of a good (or poor) match could then be used in updating the fuzzy ontology. In terms of the overall concept, specifying and maintaining the fuzzy relationship weights is one of the key challenges, and learning based on user feedback is a solution worth exploring.

6. Summary and conclusions

Fuzzy ontologies have been proposed as a solution to the difficulties that Semantic web technologies have with addressing uncertainty and inconsistency. Our interest in fuzzy semantics, however, is based on more practical reasons. A concrete benefit of fuzzy ontologies is the extension of information queries – allowing the search to also cover related results, and make the decisions about relatedness based on modelled domain knowledge. Experts working in process industry could benefit from such an enhanced search engine, since plenty of plant knowledge is being stored but effective retrieval is still challenging. Search applications based on keyword annotations are not a new idea. In process industry, applications like electronic diaries have long included such functionality. However, ontologies do offer a way of capturing and putting to use some "common sense" from domain expertise, which is something methods like text mining have a hard time mimicking.

Indeed, it is the captured common sense that separates ontology-based solutions from generic search engines. Tools like Google that are more or less based on statistical analysis of source information are powerful in a general context, since there is no way of modelling the semantics of all information. Ontologies, by comparison, require a huge construction effort, but for a limited domain, they can enable computers to put human expert knowledge to good use.

Still, specifying and maintaining ontologies is a lot of work. Working on ontologies, we have noticed that even within the fairly limited domain of industrial process automation, ontologies constructed from different viewpoints can differ enough to warrant them unusable by other domain actors. An ontology constructed from the perspective of plant design or maintenance, for example, can seem downright illogical from the point of view of operating the plant. What someone sees as a meaningful relationship between domain concepts can be useless information in another context. Interestingly, however, one of the proposed benefits of fuzzy semantics is the more flexible mapping between different ontologies.

Whether fuzzy ontologies will have any impact depends by and large on whether efficient solutions are created for specifying and maintaining the fuzzy relationship

weights. Essentially, the set of weights is a huge amount of numbers to process manually, as we were forced to do with our demonstration application. Ways of automating the task are at this stage conceptual. Value assignment would benefit from automatic methods based on analysis of source information, and the values could be updated on the basis of user feedback on search results.

In the KNOWMOBILE project, we examined the use of fuzzy ontologies in the retrieval of stored reports of industrial plant knowledge. Since no well-established definitions or mature tools exist, we adopted a practical approach – defining those constructs that we deemed necessary in order to adequately capture domain semantics, and implementing a demonstration application in which to test those constructs against actual industrial data. The demo was evaluated with the help of industry experts. Although the extended query tool seems promising, there was a great deal of worry about the amount of work needed for maintaining the fuzzy ontology relationship weights.

Practical lessons learned from the demonstration include:

- A fuzzy ontology is heavily dependent on the application domain and purpose of the tool it is developed for. This may limit its applicability to other purposes and domains.
- In addition to having their common-sense interpretation, many terms (e.g. conversion) have specific meanings in different domains. Namespaces might be needed to tell the difference.
- Furthermore, there may be the need to reuse upper-level concepts and combine domain-specific keyword sets in some software applications. The solution might be to divide the keywords into separate ontologies that can be easily integrated.

Much of the research in the KNOWMOBILE project was focused on the fuzzy ontology. However, to truly enable knowledge mobilisation, several topics should be addressed: combining information from several different sources, mapping different ontologies, supporting diverse (especially mobile) platforms, and promoting flexible and proactive user interaction, to name a few. Even for reasoning purposes, approaches other than fuzzy ontologies are surely needed, particularly in process industry, where masses of measurement data are processed. Methods such as data mining and case-based reasoning are also necessary pieces of the overall puzzle. To address the overall challenge, we must expand our view from a single reasoning mechanism to a broader concept and system architecture. Accordingly, we have been outlining an idea of a *knowledge portal* for mobilising knowledge from heterogeneous information sources to users with different contexts (Figure 32).

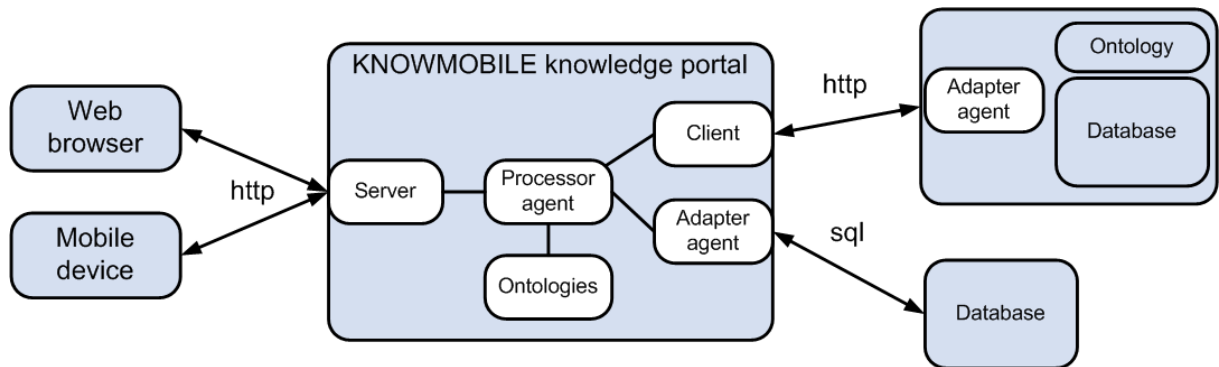


Figure 32. The concept of a knowledge portal.

In Figure 32, a portal is used to distribute knowledge to users with specific needs and contexts of use. Knowledge is extracted from a number of different sources, some of which are simple databases, while others already contain intelligent methods to process information (such as ontology-based data representation). Agent technology is used to handle translations of passed information, since a central approach could not tackle all the different representations used in the different information sources. The approach must be distributed, with reasoning capability spread out over the network, especially if we presume that ontologies or data schemas of different sources change over time. Agent technology could also provide solutions for flexible, supportive user interfacing.

References

- An, Y., Chun, S., Huang, K. & Geller, J. 2008. Assessment for ontology-supported deep web search. 10th IEEE Conference on E-Commerce and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, pp. 382–388.
- Batres, R. et al. 2007. An Upper Ontology based on ISO 15926. *Computers and Chemical Engineering* 31, pp. 519–534.
- Bordogna, G. & Pasi, G. 2000. Application of Fuzzy Set Theory to Extend Boolean Information Retrieval. In: *Soft Computing in Information Retrieval; Techniques and Applications*, Crestani, F. & Pasi, G. (Eds.), Physica-Verlag Heidelberg, 2000.
- Carlsson, C., Fullér, R. & Fedrizzi, M. 2010. Knowledge Mobilisation and Fuzzy Ontology. IAMSR Working Paper 6/2010, Åbo 2010.
- Cross, V. 2004. Fuzzy Semantic Distance Measures Between Ontological Concepts. *IEEE Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2004)*, Banff, AB, Canada, June 27–30, 2004.
- Fensel, D. 2002. Ontology-based knowledge management. *Computer*, November 2002, pp. 56–59.
- Gallova, S. 2007. Fuzzy ontology and information access on the Web. *IAENG International Journal of Computer Science*, 34:2, IJCS_34_2_11.
- Gruber, T. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, Vol. 5.
- Holi, M. & Hyvönen, E. 2005. Modeling Degrees of Conceptual Overlap in Semantic Web Ontologies. <http://www.seco.tkk.fi/publications/2005/holi-hyvonen-modeling-degrees-of-overlap-2005.pdf>.
- ISO 15926-1 2004. Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 1: Overview and fundamental principles.

- Obitko, M. & Marik, V. 2003. Adding OWL Semantics to ontologies used in multi-agent systems for manufacturing. First International Conference on Industrial Applications of Holonic and Multi-Agent systems (HoloMAS 2003), Prague, Czech Republic, September 1–3, 2003.
- Pakonen, A., Tommila, T., Pirttioja, P. & Seilonen, I. 2007. OWL Based Information Agent Services for Process Monitoring. 12th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2007), Patras, Greece, Sep 25–28, 2007.
- Parry, D. 2004. Fuzzification of a Standard Ontology to Encourage Reuse. 2004 IEEE International Conference on Information Reuse and Integration, Las Vegas, USA, 2004.
- Parry, D. 2006. Fuzzy Ontologies for Information Retrieval on the WWW, Fuzzy Logic and the Semantic Web (Capturing Intelligence). In: Sanchez, E. (Ed.), Elsevier, 2006.
- Paunonen, H., Oksanen, J., Nieminen, M., Koskinen, T. & Kovalainen, M. 2006. Collaborative Interaction in Process Control. International Encyclopedia of Human Factors and Ergonomics, 2nd Edition, Karwowski, W. (Ed.), Taylor & Francis, 2006.
- Sanchez, E. & Yamanoi, T. 2006. Fuzzy Ontologies for the Semantic Web. 7th International Conference on Flexible Query Answering Systems (FQAS 2006), Milan, Italy, June 7–10, 2006.
- Salim, K., Pulvermacher, M. & Obrst, L. 2004. Toward the Use of an Upper Ontology for U.S. Government and U.S. Military Domains: An Evaluation. MITRE technical report MTR 04B0000063, 42 p.
- Thomas, C. & Sheth, A. 2006. On the Expressiveness of the Languages for the Semantic Web – Making a Case for ‘A Little More. Fuzzy Logic and the Semantic Web (Capturing Intelligence), E. Sanchez (Ed.), Elsevier, 2006.
- W3C 2009. OWL 2 Web Ontology Language – Document Overview. <http://www.w3.org/TR/owl2-overview> [referenced 29.9.2010].
- Widyantoro, D. H. & Yen, J. 2001. A Fuzzy Ontology-based Abstract Search Engine and Its User Studies. The 10th IEEE International Conference on Fuzzy Systems, Melbourne, Australia, December 2–5, 2001.
- Viinikkala, M., Syrjälä, S. & Kuikka, S. 2006. A Maintenance Demand Analyzer – a Web Service based on a Semantic Plant Model. 4th International IEEE Conference on Industrial Informatics (INDIN 2006), Singapore, 16–18 August, 2006.

Appendix A: Publications of the KNOWMOBILE project

J. Hirvonen, T. Tommila, A. Pakonen, C. Carlsson, M. Fedrizzi & R. Fullér. Fuzzy Keyword Ontology for Annotating and Searching Event Reports. *International Conference on Knowledge Engineering and Ontology Development (KEOD 2010)*, Valencia, Spain, Oct. 25–28, 2010.

A. Pakonen, T. Tommila & J. Hirvonen. A Fuzzy Ontology Based Approach for Mobilising Industrial Plant Knowledge. *15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010)*, Bilbao, Spain, Sep. 1316, 2010.

C. Carlsson, R. Fullér & J. Mezei. An Approximate Reasoning Approach to Rank the Results of Fuzzy Queries. *International Conference on Applied Operational Research (ICAOR 2010)*, Turku, Finland, Aug. 26–27, 2010.

C. Carlsson, M. Brunelli & J. Mezei. Fuzzy Ontologies and Knowledge Mobilisation: Turning Amateurs into Wine Connoisseurs. *2010 IEEE World Congress on Computational Intelligence, FUZZ-IEEE 2010*, Barcelona, Spain, July 18–23, 2010.

C. Carlsson, M. Brunelli & J. Mezei. Fuzzy Ontology and Information Granulation: An Approach to Knowledge Mobilisation. *International Conference on Information Processing and Managements of Uncertainty in Knowledge-Based Systems (IPMU 2010)*, Dortmund, Germany, June 28–July 2, 2010.

C. Carlsson & R. Fullér. Predictive Probabilistic and Possibilistic Models Used for Risk Assessment of SLAs in Grid Computing. *International Conference on Information Processing and Managements of Uncertainty in Knowledge-Based Systems (IPMU 2010)*, Dortmund, Germany, June 28–July 2, 2010.

C. Carlsson, M. Brunelli & J. Mezei. Knowledge Mobilisation for Knowledge Whenever and Wherever Needed. *23rd Bled eConference - eTrust: Implications for the Individual, Enterprises and Society*, Bled, Slovenia, June 20–23, 2010.

C. BenMoussa. Exploiting Mobile Technologies to Build a Knowledge Mobilization Capability: A Work System-based Method. *43rd Hawaii International Conference on System Sciences*, Kauai, Hawaii, USA, Jan. 5–8, 2010.

E. Patokorpi. Situated, embodied cognition and its implications for context building in knowledge mobilisation design. *Electronic Journal of Knowledge Management*, Vol. 7, Issue 3, pp. 368–377, June 2009, Academic Publishing Limited, 2009.

C. BenMoussa. Barriers to Knowledge Management: A theoretical framework and a review of industrial cases. *International Conference on Knowledge Management and Knowledge Economy (ICKMKE 2009)*, Paris, France, June 22–26, 2009.

C. BenMoussa. Why Do We Need a New Approach to Knowledge Management? Some Answers Using Activity Theory. *International Conference on Knowledge Management and Knowledge Economy (ICKMKE 2009)*, Paris, France, June 22–26, 2009.

C. BenMoussa. Impediments to Knowledge Management. *2009 International Conference on Information Management and Engineering (ICIME 2009)*, Kuala Lumpur, Malaysia, Apr. 3–5, 2009.

E. Patokorpi & C. Carlsson. Knowledge Mobilisation in the Light of Grounded Cognition. *IADIS International Conference – Information Systems 2009*, Barcelona, Spain, Feb. 25–27, 2009.

E. Patokorpi. What could abductive reasoning contribute to human computer interaction? A technology domestication view. *PsychNology Journal*, Vol. 7, Number 1, pp. 113–131, 2009.

VTT Working Papers

- 139 Jukka Hietaniemi & Esko Mikkola. Design Fires for Fire Safety Engineering. 2010. 100 p.
- 140 Juhani Hirvonen, Eija Kaasinen, Ville Kotovirta, Jussi Lahtinen, Leena Norros, Leena Salo, Mika Timonen, Teemu Tommila, Janne Valkonen, Mark van Gils & Olli Ventä. Intelligence engineering framework. 2010. 44 p. + app. 4 p.
- 141 Juha Forström, Esa Pursiheimo, Veikko Kekkonen & Juha Honkatukia. Ydinvoimahankkeiden periaatepäätökseen liittyvät energia- ja kansantaloudelliset selvitykset. 2010. 82 s. + liitt. 29 s.
- 142 Ulf Lindqvist, Maiju Aikala, Maija Federley, Liisa Hakola, Aino Mensonen, Pertti Moilanen, Anna Viljakainen & Mikko Laukkanen. Hybrid Media in Packaging. Printelligence. 2010. 52 p. + app. 7 p.
- 143 Olavi Lehtoranta. Knowledge flows from incumbent firms to newcomers. The growth performance of innovative SMEs and services start-ups. 2010. 36 p. + app. 2 p.
- 144 Katri Grenman. The future of printed school books. 2010. 42 p.
- 145 Anders Stenberg & Hannele Holttinen. Tuulivoiman tuotantotilastot. Vuosiraportti 2009. 2010. 47 s. + liitt. 5 s.
- 146 Antti Nurmi, Tuula Hakkarainen & Ari Kevarinmäki. Palosuojattujen puurakenteiden pitkäaikaistoimivuus. 2010. 39 s. + liitt. 6 s.
- 147 Juhan Viitaniemi, Susanna Aromaa, Simo-Pekka Leino, Sauli Kiviranta & Kaj Helin. Integration of User-Centred Design and Product Development Process within a Virtual Environment. Practical case KVALIVE. 2010. 39 p.
- 149 Tommi Ekholm. Achieving cost efficiency with the 30% greenhouse gas emission reduction target of the EU. 2010. 21 p.
- 150 Sampo Soimakallio, Mikko Hongisto, Kati Koponen, Laura Sokka, Kaisa Manninen, Riina Antikainen, Karri Pasanen, Taija Sinkko & Rabbe Thun. EU:n uusiutuvien energialähteiden edistämisdirektiivin kestävyyskriteeristö. Näkemyksiä määritelmistä ja kestävyden todentamisesta. 130 s. + liitt. 7 s.
- 151 Ian Baring-Gould, Lars Tallhaug, Göran Ronsten, Robert Horbaty, René Cattin, Timo Laakso, Michael Durstewitz, Antoine Lacroix, Esa Peltola & Tomas Wallenius. Wind energy projects in cold climates. 2010. 62 p.
- 152 Timo Laakso, Ian Baring-Gould, Michael Durstewitz, Robert Horbaty, Antoine Lacroix, Esa Peltola, Göran Ronsten, Lars Tallhaug & Tomas Wallenius. State-of-the-art of wind energy in cold climates. 2010. 69 p.
- 153 Teemu Tommila, Juhani Hirvonen & Antti Pakonen. 2010. Fuzzy ontologies for retrieval of industrial knowledge - a case study. 54 p. + app. 2 p.