Title	A fuzzy ontology based approach for mobilising industrial plant knowledge
Author(s)	Pakonen, Antti; Tommila, Teemu; Hirvonen, Juhani
Citation	15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010), Bilbao, Spain, 13 - 16 Sept. 2010
Date	2010
Rights	Copyright © (2010) IEEE. Reprinted from 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010). ISBN 978-1-4244-6849-2. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of VTT Technical Research Centre of Finland's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubspermissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

A Fuzzy Ontology Based Approach for Mobilising Industrial Plant Knowledge

Antti Pakonen, Teemu Tommila, Juhani Hirvonen
VTT Technical Research Centre of Finland
P.O. Box 1000, 02044 VTT
Espoo, Finland
antti.pakonen@vtt.fi, teemu.tommila@vtt.fi, juhani.hirvonen@vtt.fi

Abstract

Semantic Web technologies – ontologies in particular – aim at efficient access to heterogeneous, distributed knowledge. However, current ontology languages such as OWL cannot properly address uncertainties, inconsistencies or contradictions. Fuzzy ontologies have been proposed to fix these shortcomings and further enhance information retrieval.

The domain of industrial process plants faces many knowledge management challenges. Knowledge in e.g. the form of written reports is stored in different systems, but retrieval is often ineffective and reuse therefore limited. This paper presents an attempt at applying a fuzzy ontology for searching reports of past situations of interest at a process plant. The aim has been to get richer search results from a knowledge base by extending the query with fuzzy neighbour concepts.

1. Introduction

Operating an industrial process plant involves handling a vast amount of information. Control systems at a paper mill, for example, continuously track thousands of measurements, intelligently control and supervise the operation and performance of a mass of complex devices, and plan and schedule production at runtime. In addition, there are more universally familiar challenges. Knowledge in the form of written reports is stored into e.g. electronic diaries, maintenance databases, or laboratory information management systems (LIMS). These documents might later provide valuable insight, for example when similar situations at the plant occur. Knowledge management efforts might enable transparent access to all relevant information, but often the information retrieval functionality leaves much to be desired.

Actually, the term "knowledge management" can already have a somewhat bad ring to it, as millions of euros have been spent building massive databases, that – although serve their purposes in fusing and storing information – ultimately are just one more system to deal with that useful knowledge can be difficult to extract from. Some researchers now prefer the term "knowledge mobilisation" [1][2] to emphasise that the ultimate need is to benefit from the knowledge in different contexts, and therefore issues like efficient, intelligent information retrieval and flexible, proactive user interfacing become essential from the start.

Ontologies are a key technology in the Semantic Web, where the intention is to provide efficient access to the vast amount of information stored in a plethora of heterogeneous languages and syntaxes on the Internet. Ontology languages such as OWL specify the semantics of domain concepts in a format understood by computers, enabling automatic reasoning over the meaning of processed information. For OWL, however, the world is black-and-white; an object, for example, either is or is not an instance of a strictly defined concept class. Fuzzy ontologies, on the other hand, are a research topic originating from fuzzy set theory. The idea is to tackle the challenges faced by "classical" or "strict" ontologies; addressing the uncertainty and inconsistency inherent in human communication.

One very practical benefit expected from fuzzy ontologies is extending the search results of information retrieval. Instead of just offering exact matches, the search can be extended to cover also related concepts. The user does not have to use the precise wording to get a useful hit, as the context of a document does not have to be exactly the same one for the user to benefit from it. As long as result flood is not an issue, we have assumed that fuzzy ontologies could have a key role in knowledge mobilisation.

Our research has been focused on validating the applicability of fuzzy ontologies in providing better information retrieval in the domain of industrial plants. This paper documents our efforts to define a fuzzy ontology in a way that still represents common-sense understanding of the meaning of the concepts at hand, in the way that "strict" ontologies do. The postulated ontology constructs have been tested with a demo application that runs extended queries against a set of actual stored industrial knowledge. Reports that are annotated with descriptive ontology keywords are queried, and a fuzzy ontology reasoner extends the query to keyword neighbours to produce both exact and fuzzy search results.

2. Industrial plant knowledge management

With the development and increase of instrumentation and ICT systems used at industrial process plants, the role of humans has changed. Controlling the plant is still done by process operators and maintenance staff, but fewer operators are now in charge of larger portions of processes. The higher degree of automation also means that their work is more than before a task of knowledge-based supervision and decision-making. Accordingly, tools for exchanging expert knowledge have become integrated functions of control systems [3].

Traditional knowledge management problems related to having heterogeneous information systems (often from different vendors) at the plant have to some extent been addressed with standardised models of factory information, such as OPC UA [4], ISO 15926 [5] or STEP [6] standards. Still, efficient use of stored knowledge remains a challenge. Work on tools supporting the exchange on process knowledge has been done since the 1990s, and tools such as process diaries are used to store organisational experience that can later be used for decision support [3]. Systems such as maintenance databases or laboratory information management systems can also serve as so-called "organisational memory", but they are often simple text applications. Intelligent methods for information retrieval have been incorporated, one example being CBR, or case-based reasoning [7]. However, the focus is often on measurement data analysis instead of written reports. There's also the chicken or the egg –dilemma of process operators unwilling to spend time on writing detailed, structured reports, if they are accustomed to not getting many useful results to their information queries.

The problem of efficient knowledge reuse is of course a universal one. Costly projects on knowledge management often solve the problem of integrating heterogeneous information from different sources, but emphasise *storing* knowledge at the cost of *utilising* knowledge. Some researchers now prefer the term "knowledge mobilisation" over the term "knowledge management" to highlight that the view on sharing knowledge should be demand-driven rather than supply-driven [1][2]. The objective is not to store information in data bases, but to activate knowledge when needed and relevant, taking into account the user context (by e.g. supporting mobile user interfacing).

3. Fuzzy ontologies

3.1. Ontologies and the Sematic Web

The aim of Semantic Web technologies is to develop data representation formats to support searching and interpretation of the heterogeneous information found in the Internet – information that is often informal and ill-structured. Traditional search mechanisms based on e.g. text mining can provide useful results, but in order to find truly relevant information, knowledge of the meaning of the exchanged content is needed. An ontology is an explicit formal specification of a shared domain conceptualisation – the objects, concepts of the domain and the relationships that exist between them [8]. Ontology languages are a key element of the Semantic Web, and provide a formal conceptual model that supports combining and reasoning over information by expressing the semantics in a machine-processable format. With languages such as OWL [9], search engines on the Semantic Web gain a basic understanding of the domain concepts and how they relate to each other.

In process industry, the need for standardised formats for knowledge exchange is also well understood, but the industry has traditionally been slow in adopting novel technologies. Real world applications tend to favour more established solutions like XML [10], but applications based on the OWL language have also been proposed (e.g. [11],[12],[13]).

3.2. Rationale for fuzzy ontologies

One of the key challenges for the Semantic Web is how to express and handle the uncertainty that is inherent in all human knowledge, and certainly with all information in the WWW. At face value, it seems that the response of the Semantic Web community is that there is no need to tackle uncertainty, and representing uncertainty at the basic level would result in structures that lack scalability [14]. The real world, however, is full of uncertainty, inconsistency, even contradictions, and so are its formalisations. Trying to define a monolithic ontology that sets a strict standard for labelling, indexing and structuring of domain knowledge for all authors of documents is ignoring one of the basic reasons for the success of the Internet [15]. An ontology that perfectly captures one persons understanding of the domain is useless for anyone else with a different view [16][11].

Fuzzy ontologies have been proposed as a solution for addressing semantics in an uncertain world. In fuzzy logic, reasoning is approximate rather than precise. Fuzzy sets are sets whose elements have a degree of membership. For example, "the class of tall men" is not a set or class in the usual mathematical sense of these terms, but a class with a continuum of grades of membership [17]. Accordingly, fuzzy ontologies reject the black-and-white reality of "classic" conceptual formalisations like OWL. The aim is to facilitate semantic interoperability by discovering semantically appropriate mappings [18] between different and independent ontologies.

As fuzzy ontologies draw influence from fuzzy set theory, a certain term can have many different meanings, each with an assigned membership value. A fuzzy mapping enables the task of finding knowledge from systems with inconsistent or even conflicting notions on domain vocabulary [14]. Information retrieval via a fuzzy ontology is characterised with great flexibility in representing not only the document contents, but also the information needs [19] – a practical example of such flexibility is the extension of information queries. A search engine based on a fuzzy ontology can expand the query by taking into account related query terms (from different ontologies) that most likely have the same meaning as those input by the user [15][20].

3.3. Defining fuzzy ontologies

Definitions for a fuzzy ontology found from the literature (e.g. [15],[20],[21]) draw influence from both fuzzy set theory and existing ontology languages. As an example, a generally understandable definition by a fairly oft-quoted author David Parry is introduced here.

According to [15], a fuzzy ontology is based around the idea that each concept (or term) is related to every other concept in the ontology, with a degree of membership assigned to that relationship based on fuzzy logic. The fuzzy membership value μ ranges from 0 to 1, and for each concept:

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

where n is the number of relations a particular concept has, which is one less than the total number of concepts in the ontology. The rule is not commutative; the membership value for the relation of concept A to concept B, μ_{AB}, is not necessarily the same as it is for the relation of concept B to concept A, μ_{BA}. The definition is illustrated in [15] with an example of how the concept “Apple” can simultaneously be understood to represent a computer company, a fruit, and a tree. Handling such ambiguity is useful for a universal search engine.

Another example of a fuzzy ontology definition, one that is more explicit in terms of different ontological constructs, can be found in [21].

Since there is no universally accepted definition of a fuzzy ontology yet, we set about defining a set of requirements for a fuzzy ontology that illustrate what kind of ontology constructs are needed in order for an ontology to be of good use in our domain of process control. Our aim was not to come up with a formal definition for a fuzzy ontology, but to experiment with constructs we have deemed useful.

4. Fuzzy ontologies for industrial plants

4.1. Requirements for ontologies addressing industrial plant knowledge

Our interest in fuzzy ontologies stems from the assumption that a search engine based on a fuzzy ontology could enhance the information retrieval of industrial plant related knowledge by extending queries to related (fuzzy neighbour) concepts. Naturally we want the exact answers, but results that are *more or less* what we asked for could also be useful. It is this uncertainty we wish to gain from fuzzy ontologies.

The interest in utilising ontologies in general is a result of the need to capture common sense knowledge to support information queries. Generic search engines like Google do not require the domain modelling effort that ontologies do, and can provide useful results just based on statistical analysis of the source text. Little is however offered in terms of actually describing the semantics, the actual meaning and the relationships between the domain concepts. For the Internet in general, that is quite a reasonable starting point. The WWW is the domain of operation for many authors on fuzzy ontologies as well, and focus is therefore on the ability to deal with different, even conflicting semantics. Our domain vocabulary, however, is fairly limited, and the relationships between concepts fairly well understood. Our foremost challenge is not just coping with all the information, but providing search results that actually make some sense.

So, to get some practical “common sense” to the fuzzy search results, it was clear from the start that many constructs from classical, “strict” ontologies were needed. We began by looking at the structure of OWL and other ontology languages, and selecting the relationship types that we considered necessary to include:

To classify concepts one needs **taxonomies** – hierarchical trees based on concept subtypes. As a practical example, the search engine needs to understand that a query for a “process component” should also cover its subclasses, e.g. “pumps”, “valves” and “pipes”. In OWL, taxonomical relationships are defined with the *rdfs:subClassOf* property. For expressing taxonomies, a fuzzy ontology is of particular interest, since it enables multiple inheritance, and therefore is not effected by the so-called “diamond problem” of semantic ambiguity. For example, we can very well define that the concept of “holes” (in the produced paper) is a subclass of both “operational problem” (caused by poor management of plant functions) and “quality problem” (an unacceptable material property), and a fuzzy ontology will compute.

An industrial plant is composed of physical systems of mechanical and electrical equipment, and each system consists of subsystems and finally of atomic components at the lowest level of decomposition. Those systems are there to perform different functions in order to accomplish the overall production task, e.g. paper making. These functions can as well be broken down to their parts. The search engine needs to understand that problems related to e.g. the function of water removal, being a part of the paper forming function, are also relevant in the wider context, so **partonomies** are needed to express these compositions. Specifically, fuzzy partonomies are needed, since the boundaries of systems and functions are often all but clear. A headbox, for example, can be considered a part of both the paper machine, and the stock preparation section of a paper mill (see Figure 2). OWL does not include a built-in partOf property, but such relationships can be defined with a *ObjectProperty* of the type *TransitiveProperty* (partOf(x,y) and partOf(y,z) implies partOf(x,z)) [9].

Finally, there are functional and other **dependencies** that link concepts together, for example: 1) Physical processes are linked to certain activities – the "wet end" subprocess performs the act of "forming", removing water of the pulp stock to form a fabric. 2) Certain materials are also linked to activities – process chemicals like "biocides" and "polymers" are used for functions like "retenting", a type of filtration process, which in turn is closely associated with process variables like "pH". In OWL, such dependencies between classes are defined with *ObjectProperty* relations that, like classes, can be arranged in subclass hierarchies. These dependencies capture common sense logic of how concepts relate to each other. The challenge in terms of modelling is to find a suitable level of precision – too generic a model is of little use, while too specific a model is difficult to maintain and reuse (a problem that has plagued the so-called expert systems).

4.2. A practical approach to industrial plant fuzzy ontology

The intended purpose of our fuzzy ontology is to extend an information retrieval query. A user interested in stored pieces of knowledge selects suitable **keywords** to define a query, and the ontology is used to select additional keywords that are related to a degree. With that in mind, the concepts of *keywords* their *relatedness* have a lot to do with how the fuzzy ontology is constructed (see Figure 1).

The keywords are divided into **keyword categories** that strive to be orthogonal. Instead of a flood of separate keywords for "brightness control", "pH control", or "dosage control" etc., we define keyword categories that separate process variables like "pH" and "brightness" from the activity of "process control". Current keyword categories are:

- **Event type** – an interesting period in the operation of a plant; it may be a problem (e.g. "instability", "web breaks") or a neutral observation (e.g. "test run") or a success story.
- **System type** – the mechanical and electrical equipment, software and people used to run the plant.
- **Function type** – activities required to accomplish the overall production task. Examples include "dosing", "mixing" and "process control".
- **Variable type** – process variables that characterise the performance or state of the plant; e.g. "pH", "conductivity", "opacity".
- **Material type** – materials and substances that have a purpose in the production chain; raw materials like "pulp", or process chemicals like "sodium hydroxide".

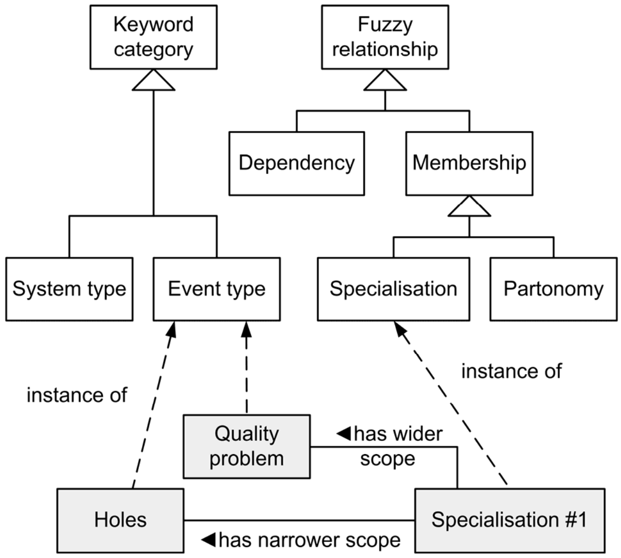

Figure 1. A simplified view of part of the OWL ontology used to represent fuzzy concept relationships.

Fuzzy relationships between keyword instances are described by a few fundamental relationship types: **specialisation** (taxonomy), **partonomy** and other **dependencies**. What makes this ontology fuzzy is that each relationship defines *a degree* of overlap of two concepts. The overlap degree ranges from 0 to 1, but can be expressed in qualitative terms such as "minor", "moderate", "significant" or "total".

However, the specialisation and partonomy relationships are not symmetric. For two keywords A and B, we have to consider their specialisation and partonomy relationship in both directions:

- **Inclusion**: The degree to which the population represented by A is included in the population represented by B.
- **Coverage**: The degree to which the members of A (the narrower scope) cover the population represented by B (the wider scope).

Let us clarify the aspects of inclusion and coverage with a few examples. In Figure 2, we present a view of the system decomposition of a part of a paper mill. We can see that the system A (headbox) is almost completely included in system B (paper machine). On the other hand, the headbox covers only a portion of the paper machine. So, the *partonomy relationship* between A and B would have the inclusion value "significant" and coverage value "moderate". For the "calendar" system, inclusion in "paper machine" would be "total", and coverage "minor".

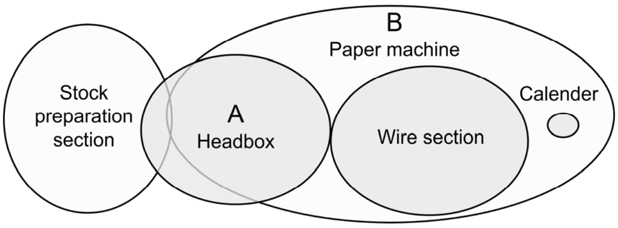

Figure 2. An example to illustrate the aspects of inclusion and coverage in the partonomy relationship between parts of physical plant equipment

From a practical viewpoint, the separation of inclusion and coverage serves a clear purpose. Fundamentally, we are interested in the relatedness of keywords in order to broaden search results. Using the keywords in Figure 2 as an example, when we are searching for information related to a "paper machine", search results involving a calender (inclusion = total) might be relevant, if a bit too specific. But, if we instead are searching for information related to a "calender", results related to a paper machine would easily be too far-fetched, so the search to that direction should be more restricted (coverage = minor).

5. A demonstration

5.1. Case scenario: accessing knowledge of past problem-solving situations

To experiment with and evaluate the feasibility of the fuzzy ontology in the industrial context, we constructed a Java-based demonstration application for conducting fuzzy queries against actual industrial report data.

The setting for our demonstration of fuzzy ontology information retrieval is that of a process chemistry expert trying to find a solution to a paper factory process control problem. Specifically, the expert is dealing with the chemistry of the wet end, part of the paper making process between pulping and wet-pressing of the paper. The idea is that the expert, in order to find a potential solution to a process control problem, wishes to look through past problem situations that are somehow (not necessarily exactly) similar to the one at hand. Often, knowledge of a similar situation in the past can provide valuable insight, even if the past situation does not deal with exactly the same process equipment, variable, function or chemical.

Although our setting is quite specific, the problem we are trying to solve is quite universal. Even in the same domain of the process plant, there are other applications like the process diary used by the plant operators that would benefit from enhanced search functionality based on an extended query.

5.2. Demo implementation

The demo application was implemented with Java. The Protégé ontology editor [22] was used to define and maintain the fuzzy ontology in OWL format. Actual problem solving reports focusing on the chemistry and process control of the wet end of a paper machine were collected from our industrial partners, and then annotated with the help of experts of those fields. The same experts have also assisted in evaluating the results provided by the fuzzy search engine. The annotation task would also necessitate a feasible software tool, but only a concept for such a tool was designed, as the demo implementation focused on the search functionality.

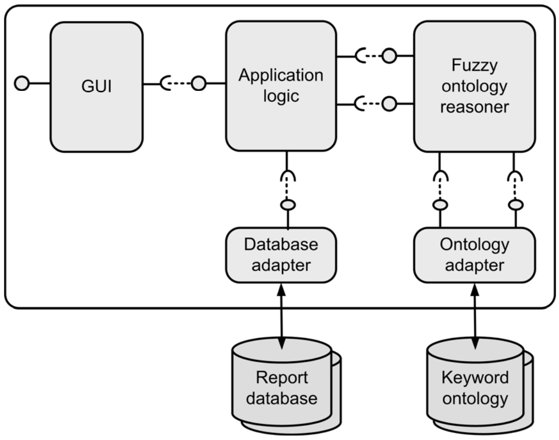

Figure 3. Demo software components.

The demo software consists of three major components (Figure 3): The "GUI" component handles the user interfacing, the "Application logic" runs the query, then processes and combines the results, and the "Fuzzy ontology reasoner" extends the original query to fuzzy neighbour concepts. Report data is accessed through a interchangeable set of adapters that use e.g. SQL or HTTP protocol. Similarly, ontology data can be read over a network.

5.3. Fuzzy report search procedure

Our procedure for running the fuzzy ontology based information query is as follows:

1. Define a query combining several keyword categories, for example: *(system = "wire section" OR "wet end") AND (function = "draining" OR "mixing") AND (event = "quality problem").*

2. For keywords in each category, find the closest keyword neighbours and their matching degrees. For example, neighbours for keyword "wet end" can include its subprocess "web forming section" or its function "forming".

3. Retrieve all the reports that have the keywords either given directly in the query or mentioned in the corresponding list of closest neighbours (extended query).

4. Calculate a “combined matching degree” for each report found by a) selecting for each category the best (MAX) matching degree between the report keywords and the extended query, and b) selecting the from all the category-based best matches the lowest match (MIN).

The extension of the query in step 2 is described in the following section.

5.4. Fuzzy keyword query extension

Within each keyword category the neighbours of a given search term are found on the basis of the specialisation and partonomy relationships. When traversing a keyword taxonomy or plant decomposition hierarchy, we use the concepts of inclusion and coverage explained above. This is done in a way that makes the relationships asymmetric. The matching value, i.e. the semantic distance from ”quality problem” to ”holes” is different from the distance from ”holes” to ”quality problem”. Furthermore, specialisations and partonomies are treated differently in the calculations. This is because these relationships have different interpretations in our ontology.

The third relationship, dependency, is mostly used to model cross-category links between keywords in different taxonomies. They are also useful for defining approximate synonyms within one keyword taxonomy. Different from specialisations and partonomies, dependencies are symmetric.

The search for keyword neighbours can be extended to any numbers of jumps in the taxonomy trees along the three relationship types. The steps are combined simply by multiplying the individual matching values of each relationship.

Parameters that affect search results – the minimum acceptable value of a keyword match, the search depth in terms of how many concept relationships the reasoning engine is allowed to jump across – can easily be tuned and experimented with in the demo application. Modification of these parameters makes it straightforward to limit the amount of search results in case there are too many, or broaden the search if the results are too few.

5.5. Demo user interface

For demo user interfacing, we developed graphical tools for query definition and result browsing, and also auxiliary tools for ontology browsing and debugging.

The demo screenshot in Figure 4 shows the dialog window for defining the query. The user can define a timeframe for results, in order to rule out report entries that are too old, or to look for entries from a certain period of interest. Then, query keywords are selected from categories such as event (e.g. “instability”), system (e.g. “paper machine”), function (e.g. “water removal”), variable (e.g. “pH”) and material (e.g. “sodium hydroxide” or “packaging paper”). Several keywords for each category can be selected, or none.

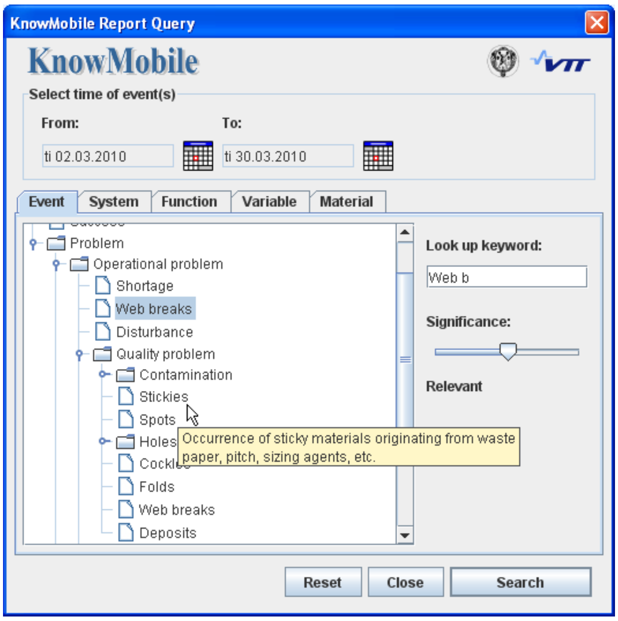

Figure 4. The demo application window for query keyword selection

The tool assists the user in finding the proper keywords 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. One of the benefits of the fuzzy ontology is the ability to process multiple inheritance. For example, in Figure 4, the keyword “Web breaks” can be found as the subclass of both “Operational problem” and “Quality problem”.

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 demo screenshot in Figure 5 shows an example of fuzzy search result. The results are listed in the order of match value, and colours are used to illustrate how well both the results and their individual keyword annotations match with the keywords used in the query.

The first result in this exemplar output 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 lessens, 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.

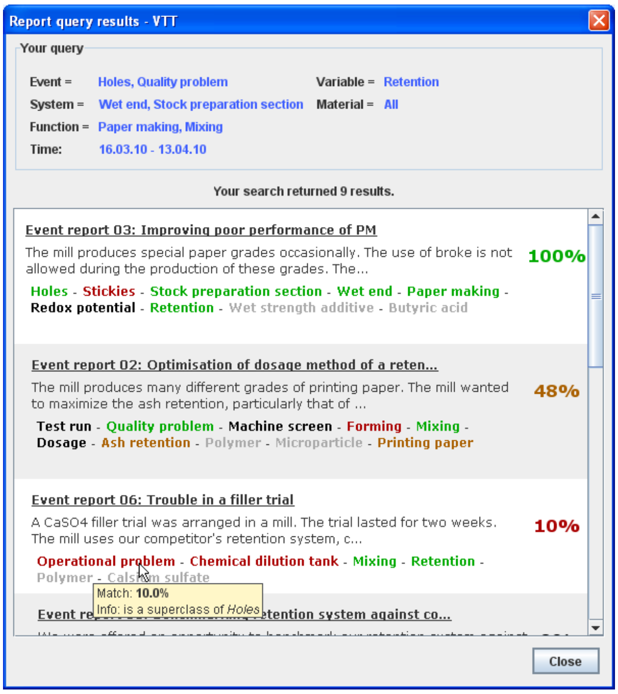

Figure 5. The demo application displaying the results of a fuzzy search

The user is also provided with some insight into the fuzzy reasoning process (Figure 5). 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 keywords (Holes) with minor (0.1) coverage.

5.6. Further issues

At the heart of the demonstration is the fuzzy ontology. The way we have specified the fuzzy constructs and the reasoning scheme can be seen as a "common sense exercise". We are currently working with our research partner, the Institute for Advanced Management Systems Research (IAMSR) of Åbo University, to test different application logics more soundly based on fuzzy set theory. One of the questions we will be looking at is whether a more theoretic approach can sufficiently capture the domain-specific knowledge included in our common-sense reasoning engine. The modular demo architecture allows us to easily experiment with different approaches.

We have used the structures of OWL and Protégé to express the fuzzy relationships, which is something those tools are not essentially made for. The lack of established languages and tools for fuzzy ontology development is a problem. But even more fundamental is the question of how to define and maintain the fuzzy relationship values, which are essentially a huge set of numbers to process manually. Value assignment would benefit from automatic processing based on e.g. text mining, and the values should be constantly updated on e.g. the basis of user feedback on search results.

6. Conclusions

Semantic Web technologies have been around for a while, but whether they will have a lasting impact on how information is processed in the Internet remains to be seen. Plenty of critique has been aimed at the basic structures behind Semantic Web languages like OWL. Fuzzy ontologies promise not only a theoretic solution to the difficulty of handling the inherent uncertainty and ambiguity of human knowledge, but also practical tools for more efficient information retrieval.

We have looked at the possibilities fuzzy ontologies can provide for knowledge mobilisation in our domain of industrial process plants. We have specified a fuzzy ontology to describe knowledge related to a paper mill, and implemented a demo tool for running extended queries against stored reports of knowledge. Since no established languages for fuzzy ontology specifications (and therefore no established software tools for fuzzy ontology reasoning) exist yet, we took a practical approach, and defined the constructs we deemed necessary for capturing sufficient knowledge in our domain, and experimented with those constructs in a demonstration application with a real-world industrial context. The search results provided by the demo seem promising and systematic testing and evaluation with industry experts is underway. Our objective was not to come up with a formal definition of a fuzzy ontology, but to define and experiment with ontology constructs we deemed useful.

One of the key issues regarding the feasibility of a fuzzy ontology is how the fuzzy membership values should be assigned and maintained. Although experts might need a way to directly modify and fine-tune the concept relationship parameters (upon noticing unfitting search results, for example), one cannot expect that the maintenance of the values is a task done manually. Processing based at least partly on automatic means (e.g. text mining) is needed.

Efficient reuse and maintenance of domain ontologies is always a challenge, even without fuzzy set theory to complicate things further. Often even ontologies developed for the exact same, restricted domain but from a different viewpoint (e.g. process plant operation vs. process plant maintenance) may be of limited value [11][16]. Generic search engines like Google that are more or less based on statistical analysis of source information do not require the kind of effort that the construction, use and maintenance of useful ontologies does. The trade-off, however, is that an ontology can capture human expert knowledge in a way computers can put to good use. Search results based on an ontology can actually "make sense", because common sense understanding of the meaning of the concepts and their many relationships is employed.

Knowledge mobilisation is a term used to describe research activities that aim at avoiding the pitfalls of many knowledge management projects. The objective is not only to store knowledge, but to put it to good use. Fuzzy ontologies could be a piece in the puzzle along with other methods such as case-based reasoning, but novel ideas are also needed for user interfacing and system architectures.

Acknowledgements

The research project KNOWMOBILE is a joint venture of the Institute for Advanced Management Systems Research (IAMSR) of Åbo University, and VTT Technical Research Centre of Finland. Our research has been funded by Tekes (Finnish Funding Agency for Technology and Innovation), and industrial companies Metso Automation, Kemira, Ruukki 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.

References

[1] 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, January 5-8, 2010.

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

[3] H. Paunonen, J. Oksanen, M. Nieminen, T. Koskinen, M. Kovalainen, "Collaborative Interaction in Process Control", *International Encyclopedia of Human Factors and Ergonomics*, 2nd Edition, Karwowski, W. (Ed.), Taylor & Francis, 2006.

[4] OPC Unified Architecture, www.opcfoundation.org/ua/

[5] ISO 15926 Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities, ISO Standard.

[6] ISO 10303 Industrial automation systems and integration - Product data representation and exchange (STEP), ISO Standard.

[7] I. Penttinen, O. Saarela, M. Nappa, K. Luomi, O. Rimpinen, "Knowledge-based reasoning about paper machine fabrics", *Papermaking Research Symposium (PRS 2009)*, Kuopio, Finland, June 1-4, 2009.

[8] T. Gruber, "A Translation Approach to Portable Ontology Specifications", *Knowledge Acquisition*, Vol 5., 1993.

[9] OWL Web Ontology Language, W3C Recommendation, http://www.w3c.org/TR/owl-features/, 2004.

[10] M. Paljakka, J. Söderman, M. Karaila, T. Syrjänen, M. Uusitalo, H. T. Heikkilä, "SEFRAM Service Framework for Process Industry Information Exchange", *Automaatio XVIII Seminaari*, Helsinki, March 17-18, 2009

[11] A. Pakonen, T. Tommila, T. Pirttioja, I. Seilonen, "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.

[12] M. Viinikkala, S. Syrjälä, S. Kuikka, "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.

[13] M. Obitko, V. Marik, "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.

[14] C. Thomas, A. Sheth, "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.

[15] D. Parry, "Fuzzy Ontologies for Information Retrieval on the WWW", *Fuzzy Logic and the Semantic Web (Capturing Intelligence)*, E. Sanchez (Ed.), Elsevier, 2006.

[16] D. Parry, "Fuzzification of a Standard Ontology to Encourage Reuse", *2004 IEEE International Conference on Information Reuse and Integration*, Las Vegas, USA, 2004.

[17] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, Vol. 8., Elsevier, 1965.

[18] V. Cross, "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.

[19] G. Bordogna, G. Pasi, "Application of Fuzzy Set Theory to Extend Boolean Information Retrieval", *Soft Computing in Information Retrieval; Techniques and Applications*, F. Crestani, G. Pasi (Eds.), Physica-Verlag Heidelberg, 2000.

[20] D. H. Widyantoro, J. Yen, "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.

[21] E. Sanchez, T. Yamanoi, "Fuzzy Ontologies for the Semantic Web", *7th International Conference on Flexible Query Answering Systems (FQAS 2006)*, Milan, Italy, June 7-10, 2006

[22] The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/