

# Experiences with Ontology Development for Value-Added Publishing

Maia Hristozova

Department of Computer Science and Software  
Engineering,  
The University of Melbourne,  
Australia  
majah@cs.mu.oz.au

Leon Sterling

Department of Computer Science and Software  
Engineering,  
The University of Melbourne,  
Australia  
leon@cs.mu.oz.au

## ABSTRACT

This paper presents our practical experience of developing an ontology using the EXPLODE method for Value-Added Publishing. Value-Added Publishing is a relatively new area of electronic information distribution that extends beyond the established simple modes of online publishing. VAP considers the needs of the user as a primary measure of effectiveness of an online document by evaluating various metrics such as author reputation and number of citations. At implementation level the features of VAP can be achieved by customised information extraction agents. The domain of VAP raises many challenges. VAP is being developed by different groups of researchers, the requirements for the ontology come from many different places, not necessarily consistent with each other. Since VAP is still not a strongly established field, its borders and the issues that it addresses often change. EXPLODE is a method which is suitable for a dynamic environment. In this paper we give an overview of EXPLODE and describe the issues that we encountered and the decisions we had to make regarding the ontology design, structure and content.

## 1. Introduction

Since 1997 ontology-based information extraction agents have advanced significantly, maturing in width and breadth of capabilities [6, 24, 25]. No longer is the state of the art a lone information agent treading a one-dimensional path through a field of data. Growing interest in multi-agent systems is providing a platform for realizing much more sophisticated outcomes through the interactions of numbers of information agents, as the outputs of individual agents can now be combined and manipulated. The grouping of multiple agents into a single system implies interaction between the individuals, in effect the construction of a community of software programs. It is widely acknowledged [15, 16] that without some shared or common knowledge, the member agents of such group systems have little hope of effective communication. The shared knowledge required could be common experiences, public information or an agreed set of definitions and meanings of basic communicable concepts. The term ontology is borrowed from philosophy to describe the latter, but the usage of ontologies has

become common in the AI community. Particularly in the Semantic Web [8] and agent communities, the most common way to enable software agents to communicate is to give each of them the same specified conceptualization, or ontology, of the domains they are expected to work in.

A number of attempts have been made to lay guidelines for the development of ontologies in much the same vein as the traditional approaches to large software application development; these are discussed in Section 3. However, as this paper describes, these methods are generally unsuited to the development of the kinds of ontology generally required by multi-agent systems. Within the Intelligent Agent Lab at The University of Melbourne, we have designed a lighter approach to ontology development for multi-agent systems called EXPLODE, which is described briefly in Section 4. We then detail our experiences from applying EXPLODE to the task of developing an ontology to be used by agents in a multi-agent system that implements the recent theory of Value-Added Publishing.

## 2. Value-Added Publishing in Theory and Practice

One of the most important and challenging problems in computer science is the development of effective technologies that support access to online information. With the expansion of the Internet useful tools for finding relevant published data are needed. In [7] Vannevar Bush proposed what became the area of information access and information extraction/retrieval. This area was concerned with modelling, designing and implementing systems able to provide fast and effective content-based access to a large amount of information. Later the aims of these systems were re-defined. The need to estimate the relevance of documents to the user's information needs was realised. This is a very difficult and complex task, since it is flavoured by subjectivity and must cope with vagueness and uncertainty. Further, the traditional after-publication activities such as the publication of revisions, corrections, updates and new editions need to be managed within the automated online process.

To face these challenges Hal Berghel has identified a new area of e-publishing called Value-Added Publishing [2] which is addressed in this paper. In similar way to traditional e-publishing Value-Added Publishing (VAP) acknowledges the need for distributing static and dynamic publications via the Internet. Additionally it often includes elements such as sophisticated multimedia and hypermedia technologies, secure transactions and communications and billing and charging systems (though common standards are still lacking). VAP steps beyond that and tries to recognize the specific extensions, techniques and methods that increase the usefulness of a publication and help publishers to meet their specific reader's needs.

To assist their readers in finding what they need, publishers must consider where their publications fit in the overall picture of online documents. By associating publications with subjects and topics, the sea of online documents becomes easier to navigate. Users move from one cluster of documents to another, finding documents that cover similar topics in one virtual place and so greatly reducing the time taken to find what they want. Digital documents can be firmly integrated into a cluster of related documents - a significant benefit that couldn't be achieved by the mechanisms available to traditional publishing. The central issue in such an approach is the capability of publishing to not only provide the documents, but to represent and supply their connections to other data sources, as well as other valuable information. For more information and examples of VAP please refer to [2, 3].

In the following section some of the techniques to achieve the goal of adding value to publications will be briefly introduced.

## **2.1 VAP Techniques**

In the context of VAP, where the information carriers and venues accept from and react to additional factors, the challenges are likely to be found in areas such as content enhancement, meta-data, confidence indicators and some others, all of which are described in the following section. They all contribute to some extent to the value of a publication and when combined together can significantly ease the difficult task of assessing a particular publication or comparing different publications. For brevity in this paper only few of them will be described.

### *2.1.1 Content enhancement*

Content enhancement involves "enrichment of the semantic and syntactic content of a document" [2]. The theory that lies behind this is that the value of the content is dependent on people's ability to read it, view it, use it and reference it. From the information retrieval point of view, data which cannot be found or used is worthless. Enhancement leads to attempts to extract more meaning from the documents, i.e. the semantic content could be summarized, reported or abstracted by intelligent agents, natural language processing machines, translation systems and other mechanisms. When a document is cited, an appropriate reference entry can be automatically generated or looked up.

### *2.1.2 Meta-data*

In the context of value-added publishing effective meta-data facilitates for example resource description and discovery, the management of information resources and their long-term

maintenance. In the context of digital resources, there is a wide variety of metadata formats. These range from the basic records used by robot-based Internet search services, through relatively simple formats like the Dublin Core Metadata Element Set ([www.dublincore.org](http://www.dublincore.org)) and the more detailed Text Encoding Initiative ([www.tei-c.org](http://www.tei-c.org)) header to highly specific formats like the FGDC Content Standard for Digital Geospatial Metadata, the Encoded Archival Description (EAD) and the Data Documentation Initiative (Codebook <http://www.icpsr.umich.edu/DDI>) that continually increase in complexity.

### *2.1.3 Confidence Indicators*

A practical way to provide useful information about a document or resource is to employ 'confidence indicators'. The purpose of a confidence indicator is to increase the assurance a reader can have that a particular document will be useful for them. The value of a document to a particular person may seem more a subjective issue than a technical one, but a lot of research has been done (especially by the commercial and advertising companies) in this area and it would be remiss not to describe the essential elements.

Confidence indicators work on the principle that if someone with similar interests to the user found a particular publication helpful, then it is likely to be helpful to the user as well. Generally, the size and nature of the audience of a publication is an indicator of the publication's quality or appeal.

An efficient way to guarantee to the reader that the document he/she is reading is a widely recognized one and has a value is to provide them with comments and notes from reviewers and leading authorities on the subject. Among all their other aims a function often performed by special interest groups and communities is to evaluate and recommend publications to their members.

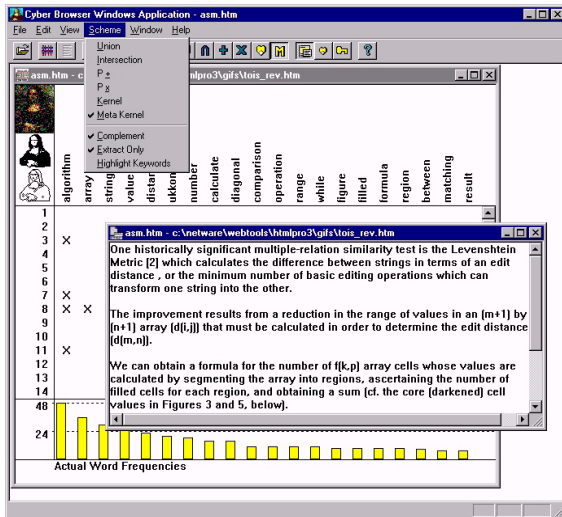
The role of intelligent agents in providing confidence indicators is as an embodiment of the reader's preferences. Personalised agents are capable of and well-suited to providing these services.

Other techniques to achieve VAP include information customisation, perceived quality of the imprimatur, word profiles, digital libraries and others, which for brevity are not described in details in this paper (see [2] for more details).

The domain of VAP raises many challenges. One such arises from the fact that VAP is being developed by different groups of researchers, not necessarily consistent with each other. Some of its characteristics have already been implemented, for example automatic summary and word profile creation (Figure 1).

Other features of VAP are still struggling with the complexity of the process. Despite this the majority of the VAP techniques are ideally suited to implementation using intelligent agents, typically personalized, mobile or information extraction. As explained in the introduction of this paper, for a group of agents to communicate effectively, an ontology is needed. It is an established practice in agent-systems research that agents need an explicit shared specification of the concepts that can be communicated within the system. For example, one agent will extract the keywords from a particular document and send them to a second agent that will perform a search on CiteSeer

(<http://citeseer.nj.nec.com>) for other papers that match the same keywords. In order to communicate both agents need to know what a 'keyword' is and additionally its relation to 'topic' for example (since some digital libraries categorize documents by topics).



**Figure 1: Keyword-oriented document extraction in Cyberbrowser [3]**

VAP is a slightly different domain than publishing and it brings some requirements for ontological commitments that are not covered in the existing ontologies for publishing, such as 'approved' or 'rated', which significantly influence the value of a particular document.

### 3. Choosing a methodology for ontology development

VAP is still not a strongly established field; its borders and the issues that it addresses often change. Thus the requirements for the ontology come from many different places, not necessarily consistent with each other - they are dynamic. This brings a need for a flexible methodology to develop the ontology, that allows change in the requirements after the initial elicitation and ensures that consistency between the requirements is maintained (through often testing) after each change. Additionally VAP is being developed by different groups of researchers and covers many different areas, sometimes not closely related to each other. The methodology must cater for the need to work on small fragments of the ontology independently as individual characteristics of VAP take shape. In these cases in order to preserve the unity of the domain frequent integration is important and so the chosen methodology must support this feature. In such a dynamic multi-agent environment as VAP, the final form of the ontology must be open for extension and change, thus the methodology chosen for its development should facilitate continuous modification of the ontology structure.

Some of the currently existing methodologies for ontology development that we have reviewed are described here.

#### 3.1 Enterprise Model Approach

In 1995 Uschold and King formulated a methodology for building ontologies by recording their experiences from developing the Enterprise Ontology [23]. The Enterprise Ontology is a collection of terms and definitions relevant to business enterprises and includes knowledge about activities and processes, organizations, strategies, marketing and more [9]. According to this methodology the phases in ontology development are:

1. Identify purpose and scope - on the basis of the available knowledge the level of formality is described.
2. Ontology capture and identification of the scope - identify the key concepts and relationships that the ontology must characterize; produce unambiguous text definitions and identify terms to refer to such concepts and relations.
3. Ontology coding - commit to a meta-ontology; choose a representation language and write the formal ontology code; integrating existing ontologies.
4. Evaluation
5. Documentation

The techniques applied during stage 2 produce a list of potential relevant concepts and erase the irrelevant afterwards. The problem that arises when creating ontologies in this way is finding the balance between extracting a large number of entries and then deleting redundant terms on the one hand, and on the other initially proposing an insufficient number of concepts and subsequently needing to extend the ontology.

#### 3.2 TOVE (Toronto Virtual Enterprise)

Developed in 1994 by [10] as a result of the experience of building an enterprise modeling ontology, the TOVE methodology involves constructing a logical model of the knowledge that is to be specified in the ontology, although the model is not built directly. Instead, competency questions are written in a formal language based on first-order predicate logic, and this formalization becomes the basis for a specification of the problem.

#### 3.3 Bernaras et alia

This methodology is described in details in [19], here only the three key steps are summarized:

1. Specification of the application.
2. Preliminary design, based on relevant top-level ontological categories.
3. Ontology refinements and structuring.

#### 3.4 KBSI IDEF5

Created by Knowledge Based Systems Inc. in 1994 the main purpose of this approach is to facilitate the design, development, modification and maintenance of ontologies [14]. This is one of the few methodologies that take into account the whole system and protocols and the main phases are:

1. Organizing and scoping: establishes the purpose, framework and the viewpoint of the ontology creation.
2. Data collection
3. Data analysis.
4. Initial ontology development

## 5. Ontology refinement and validation.

The validation tests according to this method are performed as a last stage and an essential part of the cycle is the iterative refinement and redevelopment of the structure.

### 3.5 Methontology

The METHONTOLOGY framework, as described in [4], aims to facilitate the construction of ontologies at the knowledge level. By considering three separate activities, namely the ontology development process, a proposed lifecycle and the methodology itself, the framework identifies which tasks should be performed when building ontologies, the steps to be taken to perform each task, and finally the products to be output and the means by which they are to be evaluated.

While suitable for the systems they have originally been developed for, we have identified these methodologies as inappropriate for our domain. As mentioned above VAP requires certain features of the methodology to be used, such as dynamism, the ability to change requirements during the development, frequent integration and validation. On the other hand, although specific features of VAP may evolve over time, a number of core characteristics are fundamental for the domain. Thus the methodology used to develop an ontology for VAP must facilitate establishing a constant baseline.

In view of the requirements, it is obvious that none of the above-described methodologies fully addresses these issues. Instead, we have decided to use the EXPLODE method [11] for the development of the ontology. In the following section a brief description of the EXPLODE method is presented.

### 3.6 EXPLODE

Lately much attention has been given to Agent Oriented Software Engineering [12]. Even with the emphasis on internal state and level of intelligence, agents are software modules that work autonomously in a given environment. They are pieces of software and applying principles from Software Engineering significantly contributes to the process of development, maintenance and management of multi-agent systems (MAS).

While developing ontologies, it is important to keep in mind that the final product will be a software component, a complex data structure that is to interact with the other components in the software system. Ontologies in this context they are software artifacts and they need to be treated accordingly. They are prone to the same fragility and maintenance needs that complicate software engineering, and it is entirely appropriate to apply software engineering approaches to the development of ontologies.

With all this in mind, EXPLODE was created as a method for ontology development by transferring key ideas from the eXtreme Programming methodology [1]. It is particularly suitable for dynamic and open environments thanks to its focus on immediate feedback and evaluation. Additionally the approach not only allows but favors change in the requirements at any stage of the development lifecycle. An overview of the EXPLODE method is presented in the next paragraphs.

#### *Requirements*

In the EXPLODE method requirements are determined as follows: the requirements for ontology development are extracted from both the competency questions and the system

constraints that match the specific use and application of the ontology. Competency questions are the requirements that the users of the ontology specify. They indicate the scope and content of the ontology [13].

#### *Planning*

After the competency questions and the requirements of the system are specified the process of planning is initiated. The purpose of planning is to lay out the overall ontology lifecycle including development, integration and usage. At this stage important tasks are to identify the scope and problem, to identify the concepts and relations and to consider functional as well as quality requirements. The competency questions are prioritized and decisions are made such as which question to implement at a particular iteration and what happens if two or more questions contain the same concepts.

#### *Baseline*

The baseline is a simple ontology that focuses on architectural and usage requirements. At this stage answers to difficult technical or design problems are considered.

#### *Iterations*

Iteration is the process of repeating the same development activities multiple times, generally at increasing levels of detail or accuracy. Each iteration consists of three steps – testing the competency question, iteration planning and implementation.

#### *Development*

Development is part of each iteration and consists of actual implementation of a concept or relationship and refactoring. Refactoring is an important step in the implementation process the main purpose of which is to simplify the structure of the ontology, remove redundancy, eliminate unused functionality and increase quality.

#### *Iteration Tests*

The purpose of this phase is to test if the product satisfies the requirements, in the intended environment.

#### *Acceptance tests*

Continuous integration ensures that the ontology is integrated smoothly in the system and that there are no discrepancies between the ontology structure and the agents. The purpose of the acceptance tests is to minimize these discrepancies.

#### *Maintenance*

The maintenance concept in EXPLODE is primary and the important rules to follow are the same as in eXtreme Programming, i.e. to release 'early and often'. In effect, after the first test case the rest of the process can be classified as maintenance.

## 4. Developing the Ontology

The EXPLODE method, presented in the previous section, was deployed in order to develop the ontology for the domain of VAP. The procedure that was followed and some of the major decisions made are described in the following section.

#### *Requirements*

The first step according to the EXPLODE method is to fetch the requirements both from the customers in the form of

competency questions and from the system. In our particular case the majority of the questions have been extracted from an interview with Hal Berghel during his visit at The University of Melbourne in 2002. A partial list of the competency questions is given below:

1. How many other papers on a similar topic have cited this paper in their introduction?
2. What is the ranking of the author of this publication in the most-published people website?
3. Which of these two papers is more closely related to a particular topic?
4. What is the overall confidence indicators level of this paper?
5. What is the percentage difference between two published versions written by the same set of authors?

It is important to mention that during the process of elicitation of the competency questions, the ontological engineer did not play the 'standard' role of interface between requirements and development (as in software engineering for example), but acted rather as facilitator to the customer. In this sense only partial freedom was given to the user - they provided the initial set of questions, but after some planning and discussions had occurred the user had to choose between a number of options provided by the ontological engineer.

After the competency questions had been collected and preliminary analysis was performed the requirements of the system were determined. These were extracted from the VAP features, some of which were presented earlier in this paper. A list had to be generated with all the module's input and output, corresponding to each agent achieving a particular VAP goal. The process of system requirements extraction was performed manually, and in this case the result had to be expressed in a primitive enough and simple way. In the case of a well organized MAS the system requirements can be extracted by using middle agents that serve to collect the agent's capabilities. Middle agents have been used mainly to interact between end agents [22] and they suit very well the XP principle to extract the requirements from some form of system by one module and further process them by other module. In the case of VAP the agents are mainly personalized, mobile or information retrieval. Their capabilities are specified, for example finding the awards of a paper, rating of the author, creating a paper summary, words profiles or other. The existing agents can also be providers or requesters of information (where one agent could be a provider and a requester at different times). The middle agent needs to classify the other existing agents in regards of whether they are information providers or information requesters. The task becomes even more complex if the agent receives the parameters from another agent, not from a standard input.

### *Planning*

For brevity, here we merge the details of the overall ontology planning and iteration planning. Two types of competency questions exist: core and non-core. For the domain of VAP the list of core competency questions was provided earlier. During the process of planning it was analysed in the perspective of their contribution to the ontology structure and the potential candidates for elements of the ontology were underlined. Non-core questions were provided later during the second and third

iterations of the development and thus dynamism and change of the environment were introduced and handled. An abstract of the planning process is presented below:

1. How many other papers on the similar topic have cited this paper in their introduction? - this question considers that citing a paper in the introduction brings more value to the cited paper, than if it is cited in the main body text. To answer this question, firstly the topics of the collection of papers need to be identified and then filtered to only those on the similar or the same topic.
2. What is the ranking of the author of this publication in the most-published people website? - to answer this question the author of the paper needs to be extracted and then searched on the website for the most-published people in that particular area. An extension of the question would be to ask about specific topic are, instead of searching all the published papers in all the different fields. In this case the first step is to open the website and filter the authors by category. This question almost straightforward lead to the definition of a relation or property (which one is it was identified later): author has rank.
3. Which of these two papers is more tightly related to a particular topic? - this can easily identified by extracting the word frequencies and comparing the occurrence of the main topic words. This question also leads to identifying a statement 'paper has topic'.
4. What is the overall confidence indicators level of this paper? - to answer this question the following information must be collected: awards from special interest groups and communities; awards received from professional organisations and bodies. It is clear that a 'paper has a confidence indicator' but since so far these feature of VAP has not yet been implemented the type of the confidence indicator is unknown. For the purpose of the ontology development it will be assigned as text.
5. How much is the percentage difference between two published versions written by the same set of authors? - this question identifies the difference between two papers published by the same authors but in different journals or conferences for example. The significant contribution of this question is that it identifies the level of novelty in each paper - i.e. if the same people have published very similar or identical papers at different conference, this reduces the contribution of the overall collection of publications by the same authors. For example two published papers by the same authors that have higher than 60% similarities bring less credibility than two papers published by the same authors in a completely different topics (and thus the percentage of similarity is lower than 60%). Even though this agent has been deployed in other fields (for example The University of Melbourne has a system to compare the similarities between student assignments) it has not yet been widely used by publishers. For this reason the description of the algorithm for comparison will be skipped here. The only relevant information from this question is that the expression 'set of authors' implies that the cardinality of 'author' has to be larger than 1. It also can be assumed that a 'document has version' and this version can be 'published'. Additional types of versions are extracted from the description of the capabilities of agent 3.

### *System requirements*

Some of the system requirements were mentioned in the analysis of the competency questions. Here some additional notes are included.

The use of the concept 'introduction' automatically lead to the question - what other parts can a document have? From the same question it was identified that each document contains a section where the citations are, i.e. a 'reference' section. During the first iteration the remaining parts of a document were not addressed.

The number of citations of an author was identified as an integer, which brought light how to encode 'author has number of citations'- i.e. a property of the concept 'author' rather than binary relationship between two classes.

Additionally to the third question from the information about the keyword extraction agent's capabilities it was identified that the keywords are in the form of a list, there are exactly 5, and they are strings. Also the topic of a document is one single string, usually between 1 and 5 words.

During the process of iterations it was possible to modify the existing requirements or discover and additionally include new ones.

#### *Baseline*

The baseline ontology for the domain of VAP is presented in Figure 2.

- Content (*of a publication*)
  - Abstract (*automatic summary up to 200 words*)
  - Introduction (*cites other papers*)
  - References (*cites other papers*)
- Publication (*has version in a string format*) (=document)) (*has topic*) (*has author at least 1 [ $\geq 1$  to many]*) (*has confidence indicator-text*)
  - Paper
- Person (*has name which is a string; can have a title: Prof., PhD.*)
  - Author (*writes papers*) (*has number of citations*)

**Figure 2: Baseline of the VAP ontology**

So far all the existing methodologies suggest that the ontology engineers create the baseline ontology manually. Our baseline ontology was also created manually. In a MAS though by deploying middle agents this process can be performed semi-automatically - the middle agent provides the concepts and the attributes, but the engineers define the structure. For example, based on the description of agent's capabilities, a middle agent will suggest concepts such as 'author' and 'paper', the type of the output, i.e. 'list of keywords' and maybe other information, depending on what is provided by the information extraction agent, but the relationship between them - 'writes' or 'has' - will be defined manually.

#### *Validation Tests*

The validation tests ensure the gradual development of the ontology in a step-wise refinement fashion. In the case of VAP a competency question might be: "What is the topic of this paper?". For this example, the ontology developed so far is shown in Figure 2. Clearly, at this iteration the ontology already contains sufficient concepts to fully answer the competency question. This can be determined by searching the existing ontology for the words in the competency question. At this stage the comparison was mainly syntactic, based on pattern matching, but further it could be extended to semantic mapping [18, 21] (for example by calculating the distance between the concepts according to Wordnet). As the question can be answered using the current ontology the ontological commitments are valid and the next question is tested. Pattern matching for adding concepts to ontologies has already been used by a number of other researchers and we do not consider this as an obstacle when applying it to our case.

If some words in the competency question are not found in the current ontology, it is not yet safe to assume that the entire question must be implemented. The ontology may still contain some words that are already in the ontology, and to reimplement them would cause an inconsistency. During the development of the ontology for VAP, there has not been a single core question that was found to be already answerable by the existing ontology. The only concepts that appeared constantly were 'paper', and in two of the questions, 'author' and 'topic'. For 'paper' and 'author' the plurals were identified, i.e. when the next question asked referred to 'papers', it was identified that 'paper' already exists; similarly for 'author'.

#### *Development and refactoring*

During the development of the ontology there has not been a need of significant modifications of the already existing structure. During the implementation of the second question the concept 'person' was included and after that 'people'.

In practice, there are two different approaches to ontology modification. The first one is to allow the user to identify inconsistencies in the model and change them. Modifying ontologies is covered in [17]. This sounds very easy, however from our experience we have discovered that removing classes or changing the structure (relationships) is a complex process that even intelligent agents lack capabilities to perform perfectly. In [5] is proposed a semi-automated approach for modifying ontologies, using management assistants. For the different modifications of the ontology corresponding assistants analyze the model to identify the consequences of the planned actions. The assistant then works in cooperation with the user to select and perform operations without violating the consistency of the knowledge base in order to achieve user's goal. An intelligent agent can predict the consequences of an ontology structure, thus the evaluation will be performed before the actual change. In this scenario the development of the knowledge-based agents is based on ontology reuse and development.

#### *Iteration Tests*

EXPLODE covers some of the major types of tests and following the guidelines was a straightforward process. During the development of the ontology for VAP, because of the simplicity of the ontology structure, the iteration tests were

mainly performed manually. Additionally the current version of Protégé (protege.stanford.edu) supports some redundancy tests while adding new classes.

#### Second Iteration

It was decided that during the second iteration non-core questions will be added to the structure. For example, one such question is: "What is the topic of this thesis?". During the implementation of this question the developer is alerted to the fact that the concept 'topic' is already in the ontology, and they can then add the concept 'thesis' with the relationship 'has topic'. Additionally, because the ontology already contains a concept 'paper' with a relationship 'has topic', the developer is asked if the new concept 'thesis' is related to the concept 'paper'. In this case the relations 'paper'-'topic' and 'thesis'-'topic' gives grounds for assuming a relation between 'paper' and 'thesis'. If two things are related to a common third thing, then it is quite reasonable to investigate the possibility that they have even more in common than just the third thing. If, as was the case described here, the two relationships are the same, the potential for commonality between the two things is even higher. By simple realizations such as these the produced ontology was as efficient as possible with very little redundancy.

#### Third Iteration

During the third iteration it was decided that the ontology will be extended to include synonyms. For this purpose the already implemented core and non-core questions were reviewed and reformulated with different words. During this iteration one of the problematic issues was the decision of whether to include 'document' as the same as 'paper' simply with a different name; to include it as the same as 'publication'. Subsequently 'document' was identified as the same as 'publication'.

#### Maintenance

As the domain of VAP is dynamic and researchers continually reconsider its important aspects, the hierarchical structure of the ontology is still being modified. The maintenance of the ontology consists of adding new concepts to the ontology, modification of existing concepts and deletion of concepts.

A partial list of the concepts (for readability limited to level of depth 3) of the final version is presented:

- Content (*of a publication*)
  - Abstract (*automatic summary up to 200 words*)
  - Introduction (*cites other papers*)
  - Main body
  - References (*cites other papers*)
- Library
- Publication (has Version in a Number format)
  - Paper
    - Technical Report
    - Short conference paper
    - Conference paper
  - Thesis
  - Master's
  - PhD
- Organization (quality of imprimatur in %)
  - Professional
    - ACM
    - IEEE
    - ACS
  - SIG

- Person (can have a title: Prof., PhD.
  - (could be a member of an organisation)
  - (has rating in the top 10 000 authors list <http://citeseer.nj.nec.com/allcited.html>)
  - Employee
  - Columnist
  - Editor
  - Reporter
  - Author
- Confidence Indicator
  - Readers (number of readers so far)
  - Review (could be by a SIG, an individual or a Professional Organisation) (reviews have dates)
  - Award (could be by a SIG or a Professional Organisation) (awards have dates)
  - Rating
    - Cited in Introduction by
    - Cited in References by
    - Rating of the author

## 5. Lessons Learnt and Conclusion

The lessons that we learnt during the development of the ontology were very similar to those described in [20]. We will list some additional findings:

### 5.1 Choice of methodology

As noted in Section 3 the methodology to follow when developing the ontology can have effect on the productivity and future integration. When choosing especially in multi-agent systems, features such as dynamism, flexibility, feedback, component-based development and others are very important and should influence the choice of the methodology. In our case VAP provided most of these features: with its constant development often by different groups of people (dynamic and component-based development) and need of a user feedback (confidence indicators and information customization are particularly personal and subjective measures for a value of a document).

### 5.2 Emphasis on planning

Planning might be easily overlooked, but in our case it was the stage when decisions for the whole development had to be made. For example, choosing to implement non-core questions during the second iteration and synonyms during the 3<sup>rd</sup> iteration.

### 5.3 Test different hierarchies

Different ontology engineers have different tendencies when it comes to choosing between a shallow or deep hierarchy. Since the structure can affect the speed of parsing and extracting information it is good for example to exploit different possibilities. Continuous integration and constant testing allows this to be performed without significant effort.

### 5.4 The role of Ontological Engineer

The role of OE could jump from developing the whole ontology without much user's input to the other extreme of not considering specific issues (for example choice of tools, language, depth of hierarchy and others) but simply following the user's requirements. In our case the role of the OE was

somewhere in the middle – the competency questions were reformulated by the user after some suggestions from the OE.

We also strongly felt that the suggestion to not include all the information [20] was a useful one and particularly supported by our choice of methodology.

In this paper we have described the process of developing an ontology for the domain of value-added publishing. We argued that none of the existing methodologies was applicable for our case that required dynamism, flexibility and development in stages. The chosen methodology EXPLODE was briefly introduced and applied for the development of the ontology. We have confirmed some of the lessons learnt from previous researchers when developing a lightweight ontology for a multi-agent system. We have also given some additional consideration when developing a purposive ontology.

## 6. References

- [1] Beck, K. *extreme programming eXplained: embrace change*. Addison-Wesley, Reading (2000)
- [2] Berghel, H. *Value-Added Publishing Communications of the ACM* (1999)
- [3] Berghel, H., Berleant, D., Foy, T. and McGuire, M. *Cyberbrowsing: Information Customization on the Web*. *Journal of the American Society for Information Science*, 50 (6). 505-511, (1999)
- [4] Blázquez, M., Fernández, M., García-Pinar, J.M. and Gómez-Pérez, A., *Building Ontologies at the Knowledge Level using the Ontology Design Environment*. In 11th Workshop on Knowledge Acquisition, Modeling and Management (KAW '98), (Banff, Canada, 1998).
- [5] Boicu, M. and Tecuci, G., *Ontologies and the Knowledge Acquisition Bottleneck*. In 17th International Joint Conference on Artificial Intelligence (IJCAI 01), (Seattle, America, 2001).
- [6] Borgo, S., Guarino, N., Masolo, C. and Vetere, G., *Using a Large Linguistic Ontology for Internet-Based Retrieval of Object-Oriented Components*. In *Conference on Software Engineering and Knowledge Engineering*, Knowledge Systems Institute, 528-534, (Madrid, Spain, 1997)
- [7] Bush, V. *As We May Think*. *The Atlantic Monthly*, 176 (1), (1945)
- [8] Ding, Y., Fensel, D. and Stork, H.-G. *The Semantic Web: from Concept to Percept*, eBusinessLeadership.net, (2001)
- [9] Gomez-Perez, A. *Ontological Engineering: A State of the Art Expert Update - The British Computer Society Specialist Group on Artificial Intelligence*, 33-43 (1999)
- [10] Grüninger, M. and Fox, M.S., *The Design and Evaluation of Ontologies for Enterprise Engineering*. In *Workshop on Implemented Ontologies*, European Workshop on Artificial Intelligence, (Amsterdam, Netherlands, 1994).
- [11] Hristozova, M. and Sterling, L., *An eXtreme method for developing lightweight ontologies*. In *Workshop on Ontologies in Agent Systems*, 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, (Bologna, Italy, 2002).
- [12] Tveit, A.: *A survey of Agent-Oriented Software Engineering*. NTNU Computer Science Graduate Student Conference. Norwegian University of Science and Technology, May (2001).
- [13] Kim, H.M., Fox, M.S. and Grüninger, M. *An ontology for quality management — enabling quality problem identification and tracing*. *BT Technology*, 17 (4). 131-140, (1999)
- [14] Knowledge Based Systems Inc. (KBSI). *IDEF5 Ontology Description Capture Overview*, (Knowledge Based Systems Inc., 2000)
- [15] Lister, K. and Sterling, L., *Agents in a Multi-Cultural World: Towards Ontological Reconciliation*. In 14th Australian Joint Conference on Artificial Intelligence, (Adelaide, Australia, 2001).
- [16] Lister, K. and Sterling, L. *Reconciling Ontological Differences for Intelligent Agents*. in Bouquet, P. ed. *Meaning Negotiation*, AAAI Press (Menlo Park, America, 2002)
- [17] McGuinness, D., Fikes, R., Rice, J. and Wilder, S., *An Environment for Merging and Testing Large Ontologies*. In 7th International Conference on Principles of Knowledge Representation and Reasoning (KR2000), (Breckenridge, America, 2000).
- [18] McGuinness, D.L., *Conceptual Modeling for Distributed Ontology Environments*. In 8th International Conference on Conceptual Structures Logical, Linguistic and Computational Issues (ICCS 2000), (Darmstadt, Germany, 2000).
- [19] Ostermayer, R., Meis, E., Bernaras, A. and Laresgoiti, I. *Guidelines on Domain Ontology Building*, Deliverable DO1c.2, KACTUS ESPRIT Project 8145 (1996)
- [20] Sayers, C. and Letsinger, R., *An ontology for publishing and scheduling events and the lessons learned in developing it*. In *Workshop on Ontologies in Agent Systems*, 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), (Bologna, Italy, 2002).
- [21] Stoffel, K., Taylor, M. and Hendler, J., *Efficient Management of Very Large Ontologies*. In *American Association for Artificial Intelligence Conference (AAAI-97)*, AAAI/MIT Press, (1997)
- [22] Sycara, K., *Multi-Agent Infrastructure, Agent Discovery, Middle Agents for Web Services and Interoperation*. In 3rd European Agent Systems Summer School (ACAI-01), (Prague, Czech Republic, 2001)
- [23] Uschold, M. and King, M., *Towards a methodology for building ontologies*. In *Workshop on Basic Ontological Issues in Knowledge Sharing*, IJCAI-95, (Montreal, Canada, 1995)
- [24] van Heist, G., Schreiber, A.T. and Wielinga, B.J. *Using explicit ontologies in KBS development*. *International Journal of Human-Computer Studies*, 45 (1997)
- [25] Weinstein, P. and Birmingham, W., *Service classification in a proto-organic society of agents*. In *Workshop on Artificial Intelligence in Digital Libraries*, 15th International Joint Conference on Artificial Intelligence (IJCAI 97), (Nagoya, Japan, 1997)