A proposal for a Unified Process for ONtology building: UPON[1]

Antonio De Nicola[*], Michele Missikoff[*], Roberto Navigli[**]

[*] Istituto di Analisi dei Sistemi ed Informatica
Consiglio Nazionale delle Ricerche
Viale Manzoni, 30 – 00185 Roma
{ denicola, missikoff }@iasi.cnr.it
** Dipartimento di Informatica
Università di Roma "La Sapienza"
Via Salaria, 113 – 00198 Roma
navigli@di.uniroma1.it

Abstract. Ontologies are the backbone of the Semantic Web, a semantic-aware version of the World Wide Web. To the end of making available large-scale, high quality domain ontologies, effective and usable methodologies are needed to facilitate the process of Ontology Building. Many of the methods proposed so far only partly refer to well-known and widely used standards from other areas, like software engineering and knowledge representation. In this paper we present UPON, a methodology for ontology building derived from the Unified Software Development Process. A comparative evaluation with other methodologies, as well as the results of its adoption in the context of the Athena Integrated Project, are also discussed.

1 Introduction

Ontologies, i.e. semantic structures encoding concepts, relations and axioms of a given domain, are the backbone of the Semantic Web (Berners-Lee et al., 2001), a semantic-aware version of the World Wide Web. Ontologies allow the web resources to be semantically enriched. This is a pre-condition to provide new, advanced services over the web, such as the semantic search and retrieval of web resources.

Unfortunately the community has not yet reached a consensus on one or more standard methods for building large-scale ontologies. For this reason, in this paper we propose a method derived from a well-established software engineering process, the Unified Software Development Process (Jacobson et al., 1999).

Along this line, we present UPON, a novel approach to large-scale ontology building that takes advantage of the Unified Process (UP). As a result, on one side, the adoption of the UP and the Unified Modeling Language (UML) makes ontology building an easier task for modellers familiar with these techniques. On the other side,

[1] This work is partially supported by the Interop NoE and Athena IP, 6th European Union Framework Programme.

we show how well each phase of ontology building fits in the UP, thus guiding the process of ontology development through a number of consolidated steps.

UPON is aimed at supporting the work of the ontology engineers, that we classify as knowledge engineers (KE) and domain experts (DE). Even though automatic ontology learning methods allow ontology engineers to significantly speed up the ontology building process, the automatically generated ontology always requires an additional manual validation and integration. Therefore, a manual procedure is still necessary to guide the process of releasing the final ontology.

The paper is organized as follows: in Section 2 we present our approach to ontology building. Section 3 discusses previous work in this area and provides a two-fold evaluation of UPON, the first by comparison with others methodologies, and the second in the context of the Athena Integrated Project[2]. In particular, using UPON, an eProcurement ontology was built with the support of AIDIMA[3], a research and development association, dedicated to technology and innovation transfer to the Spanish woodworking and furniture sector. Finally, in Section 4 we provide conclusions and future work.

2 UPON: UNIFIED PROCESS FOR ONTOLOGY BUILDING

In this section we present UPON (Unified Process for ONtology building), an incremental methodology for ontology building. The process we propose stems its characteristics from the Software Development Unified Process, one of the most widespread and accepted methods in the software engineering community, and uses the Unified Modeling Language (UML) to support the preparation of all the blueprints of the ontology project. UML has been already shown to be suitable to this end (Guizzardi et al., 2002), confirming its nature of rich and extensible language.

What distinguishes the UP and UPON from the other processes, respectively for software and ontology engineering, is their *use-case driven*, *iterative and incremental* nature.

UPON is *use-case driven* in that it aims at producing an ontology with the purpose of serving its users, both humans and automated systems (e.g. semantic web services, intelligent agents, etc.). User interactions take place through *use cases* that drive the exploration of all aspects of the ontology.

The nature of the process is *iterative* because each activity is repeated possibly concentrating on different parts of the ontology being developed, but also *incremental*, since at each cycle the ontology is further detailed and extended.

Following the UP, in UPON we have cycles, phases, iterations and workflows. Each cycle consists of four phases (*inception*, *elaboration*, *construction* and *transition*) and results in the release of a new version of the ontology. Each phase is further subdivided into iterations. During each iteration, five workflows (described in the next subsections) take place: *requirements*, *analysis*, *design*, *implementation* and *test*.

[2] "Advanced Technologies for Interoperability of Heterogeneous Networks and their Application", Integrated Project 507849, 6th EU FP - http://www.athena-ip.org

[3] http://www.aidima.es/aidima

Workflows and phases are orthogonal in that the contribution of each workflow to an iteration of a phase can be more or less significant: early phases are mostly concerned with establishing the requirements (identifying the domain, scoping the ontology, etc.), whereas later iterations result in additive increments that eventually bring to the final release of the ontology (Fig. 1). Notice that, as illustrated in the figure, more than one iteration may be required to complete each of the four phases. This scheme follows faithfully the Unified Process. In addition, as shown in the figure, the domain expert provides his contribution in the early workflows and partially during the *Test* while the knowledge engineer is mostly focused on the *Design* and *Implementation*.

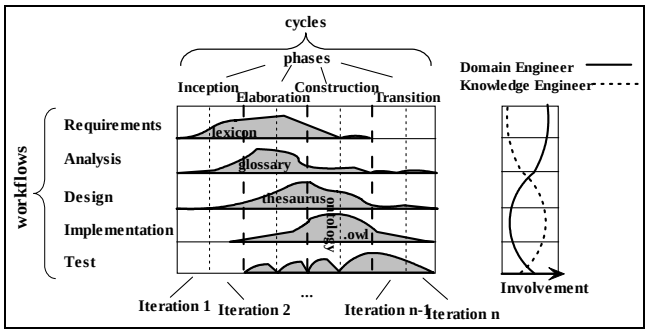


Fig. 1. The UPON Framework.

The first iterations (*inception phase*) are mostly concerned with capturing requirements and partly performing some conceptual analysis. Neither implementation nor test is performed. During subsequent iterations (belonging to the *elaboration phase*) analysis is performed and the fundamental concepts are identified and loosely structured. This may require some design effort and it is also possible that the modellers provide a preliminary implementation in order to have a small skeletal blueprint of the ontology, but most of the design and implementation workflows pervade iterations in the *construction phase*. Here some additional analysis could be still required aiming at identifying concepts to be further added to the ontology. During the final iterations (*transition phase*), testing is heavily performed and the ontology is eventually released.

The incremental nature of UPON requires first the identification of the relevant terms in the domain, gathered in a lexicon; then the latter is progressively enriched with definitions, yielding a glossary; adding to it the basic ontological relationships allows a thesaurus to be produced, until further enrichments and a final formalization produces the sought reference ontology.

In the following subsections each ontology workflow is described in detail, with the help of a practical example.

2.1 The Requirements Workflow

Requirements capture is the process of specifying the semantic needs and knowledge to be encoded in the ontology. The essential purpose of this workflow is to reach an

agreement between the modellers, the knowledge engineers, and the final users (Jacobson et al., 1999), represented by the domain experts.

During the first meetings, knowledge engineers and domain experts establish the guidelines for building the ontology. The first goal is the identification of the objectives of the ontology users. To this end, it is necessary to: (i) determining the domain of interest and the scope, and (ii) defining the purpose. These objectives are achieved by: (iii) writing a storyboard, (iv) creating an application lexicon, (v) identifying the competency questions, and (vi) the related use cases.

(i) Determining the domain of interest and the scope. Delimiting the domain of interest is a fundamental step to be performed (Uschold and King, 1995), aiming at focusing on the appropriate fragment of reality to be modelled. If the domain is huge, one or more sub-domains may also be determined.

The domain we used to validate the UPON methodology is *eBusiness*. In particular, we focused on *eProcurement*, the *business-to-business* (*B2B*) purchase and sale of goods and services over the Internet.

Defining the scope of the ontology consists in the identification of the most important concepts to be represented, their characteristics and granularity. Defining a scope means making a set of ontological commitments, bringing some part of the domain into focus at the (required and expected) expense of blurring other parts. These ontological commitments are not incidental: they provide a guidance in deciding what aspects of the domain are relevant and what to ignore.

Following Guarino et al. (1994), the ontological commitment can be seen as "a mapping between a language and something which can be called an ontology". This allows one to preliminarily identify terms as representatives of ontology concepts.

Usually at this stage modellers have only a vague idea of the role each concept will play, i.e., their semantic interconnections, within the ontology. If necessary, they can annotate these ideas for further development during subsequent iterations.

In the *eProcurement application*, the ontology chiefly concerns all the processes and the interactions between a buyer and a supplier (e.g., exchange of business documents like an *invoice* or a *purchase order*).

(ii) Defining the purpose (or motivating scenario). The reason for a new ontology, its intended uses, and the kinds of users must be established. In the *eProcurement application*, the goal of the ontology is to provide a better understanding of the domain of interest and be a support for semantic interoperability between two legacy systems. In particular, we envisage three basic uses of the developed ontology:

- search and retrieval of semantically enriched documents;
- ontology-based reconciliation of data messages exchanged between a buyer and a supplier in business transactions;
- ontology-based reconciliation of business processes between two different business partners (e.g. the steps in a purchasing activity).

(iii) Writing a storyboard. In this step the domain expert is asked to write a panel or series of panels of rough sketches outlining the sequence of all the activities that defines a particular scenario. This storyboard can be also used to extract the terminology of the domain expert.

(iv) Creating the *application lexicon*. This task can be supported by using some automatic tools to extract knowledge from documents, such as OntoLearn (Navigli and Velardi, 2004).

(v) Identifying the competency questions. Competency questions are questions an ontology must be able to answer (Gruninger and Fox, 1995). They are identified through interviews with domain experts, brainstorming, an analysis of the document base concerning the domain, etc. The questions do not generate ontological commitments, but are used during the test workflow to evaluate the ontological commitments that have been made. The usage of competency questions is more appropriate when the ontology will be used for querying and discovering resources rather than for reconciliation.

(vi) Use-case identification and prioritization. UPON proposes to take competency questions into account through use-case models. A *use-case model* serves as a basis to reach an agreement between the users (i.e., who require the ontology) and the modellers, and contains a number of *use cases*. In the context of ontologies, use cases correspond to knowledge paths through the ontology to be followed for answering one or more competency questions. Although they are to be specified during the analysis and design workflows, it is necessary to *prioritize* and *package* (i.e. group) them during requirements. The result will help dictate which use cases the team should focus on during early iterations, and which ones can be postponed.

The outcome of the Requirements Workflow is a set of documents, including those resulting from the above steps, to be extended during subsequent iterations.

2.2 The Analysis Workflow

The conceptual analysis consists of the refinement and structuring of the ontology requirements identified in previous section. The ontological commitments derived from the definition of scope are extended, by reusing existing resources and through concept refinement.

Considering reuse of existing resources: identification of relevant terms (domain lexicon). The description of this activity adheres to the view of linguistic ontology (Gómez-Pérez et al., 2004) in which concepts, at least the lower and intermediate levels, are anchored to texts, i.e. they have a counterpart in natural language.

Reuse concerns internal legacy resources as well as external resources requiring possible refinements and extensions, like interviews, documents, standards, glossaries, thesauri, computational lexicons and available ontologies.

In the *eProcurement application* the domain experts considered the following *eBusiness* standards: ebXML (http://www.ebxml.org), RosettaNET (http://www.rosettanet.org), and OAGIS (http://www.openapplications.org). The analysis of these standards comprises 2614 elements (140 from ebXML, 1873 from RosettaNET, 600 from OAGIS). A statistical analysis was done in a corpus of documents of reference to identify frequently used terms to be included in the domain lexicon. The domain experts decided to include, in this lexicon, all the terms present in at least two standards (e.g. Price, Currency, TransportMode, etc.). Other terms

were included only after approval from a wider panel of experts. After this activity, the domain lexicon contained 83 terms.

Modelling the application scenario using UML diagrams. The goal of this activity is to model the application scenario and better specify the Use Case Diagrams, drawn in the requirement workflow, with the aid of Activity and Class Diagrams. The reason to use UML is that it represents the scenario in a shared language, that allows domain experts (especially business people) to perform this activity without the support of the knowledge engineer.

Building the glossary. A first version of a glossary of concepts of the domain of interest has to be built merging the application lexicon (from the domain experts) and the domain lexicon (from the existing resources). Considering the scope of the two lexicons we can organize all the concepts in two major areas: the intersection area and the disjoint area (see Fig. 2). As done with the analysis of existing resources, it is possible to use a similar "inclusion policy": the glossary should include all the concepts coming from the intersection area and, after the domain experts approval, the ones from the disjoint area. Then domain experts should agree on the definition of concepts. It is very important that the concepts are defined according to precise references or mentioning the author of that definition.

2.3 The Design Workflow

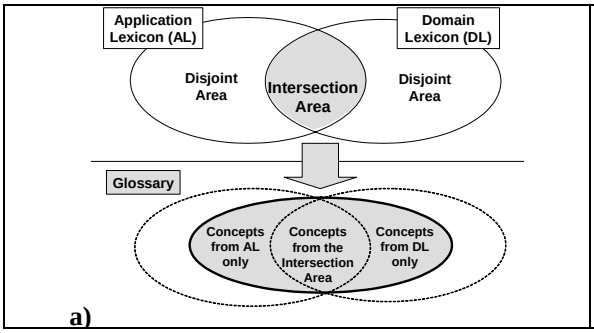
The refinement of entities, actors and processes identified in the analysis workflow, as well as the identification of their relationships, is performed during the design workflow. The design of the ontology follows the OPAL methodology (Missikoff and Taglino, 2002).

Categorising the concepts according to the *OPAL* methodology (Actor, Process, Object). Each concept can be further enriched with the identification of a top-level "category" for the defined concept (e.g. *entity* for Product, *process* for Purchase Order Issuing, *actor* for Purchasing Unit, etc.).

These "categories" include the major ontological categories, according to proposals of top ontologies, such as (Sowa, 1999), or meta-ontologies (Uschold and King, 1995). We adopted the *OPAL* methodology.

Refining the concepts and their relations. At this stage, the gradual and incremental passage from terms to concepts is made clear by the formal definition of relations between sets of synonyms identified in the previous phase.

As a first structuring step, concepts can be organized in a taxonomic hierarchy through *generalization* (the *kind-of* or *is-a* relation). Three main approaches are known in the literature (Uschold and Gruninger, 1996): *top-down* (from general to particular), *bottom-up* (from particular to general) and *middle-out* (or *combined*). The combined approach consists in finding the salient concepts and then generalizing and specializing them. This approach is considered to be the most effective because concepts "in the middle" tend to be more informative about the domain.



Purchase Order: A printed or typed document, issued by the Buyer Purchasing Unit as a firm and formal request to a specific Manufacturer/Supplier to produce and supply goods/services according to Price, Terms and Conditions previously agreed and approved. (Merriam Webster + AIDIMA)

b)

Fig. 2. a) Activity of glossary building. **b)** One of the concepts included in the *eProcurement* glossary with its definition.

The resulting taxonomy can finally be extended with other relations, i.e., p*art-of* and *association*. The outcome of this step is a UML *class diagram*, using *generalization* (*IsA*), *aggregation* (*Part-Of*) and *association* relations. A UML association relation can be labeled with a predicate and allows to represent all the relations needed for the ontology to be built.

2.4 The Implementation Workflow

The purpose of this workflow is to formalize the ontology in a language and to implement it in terms of components. Components implement concepts from the design workflow and follow the established grouping into packages (i.e. ontology portions). *Use-case prioritization* from the requirements workflow and *packaging* from all the previous workflows allow component engineers to work on different parts of the ontology to be integrated at subsequent iterations.

Components can be written down in a variety of languages and notations. The adoption of a certain formalism is appropriate as long as it conveys the appropriate expressiveness and it allows an easy reuse within the community. As a result of a long standardization effort, the Ontology Web Language (OWL: http://www.w3.org/TR/owl-features) is the main candidate for encoding an ontology to be used on the Semantic Web. The outcome of this workflow is the implementation model, including packages of implemented components.

For instance, in the *eProcurement* domain, concepts can be packaged in two groups: the ones concerning internal activities, performed inside a business organization (e.g. *Purchase Requisition Form*, *Evaluating Purchase Request*, ...), and the ones concerning interaction activities, performed between two different business organizations (e.g. *Purchase Order*, *Issuing Purchase Order*).

2.5 The Test Workflow

The test workflow allows to verify that the ontology correctly implements its requirements. UPON envisages two kinds of test. The first concerns the coverage of the

ontology over the application domain. In particular, the domain expert is asked to semantically annotate the UML diagrams, representing the application scenario, with the ontology concepts. This test is more appropriate for ontologies to be used for the ontology-based reconciliation of messages and business processes. The second concerns the competency questions and the possibility to answer them by using concepts in the ontology. For instance, in the *eProcurement application* such a test gives a positive result, since it is possible to answer to the question "*What are the documents that a company receives before a Purchase Order?*" using the ontology concepts *Request For Quotation*, *Processing RFQ*, *Sending RFQ*. This test is more appropriate for ontologies to be used for discovery and search of web resources.

3 Related Work & Evaluation

The first contributions to ontology building methods are due to Gruber (1993), Gruninger and Fox (1995), Uschold and King (1995), Uschold and Gruninger (1996), constituting the basis for many subsequent proposals. Gruber's seminal work discusses some basic ontology design criteria (clarity, coherence, extendibility, minimal encoding bias and ontological commitment). Gruninger and Fox (1995) provide a skeletal methodology for ontology building, while a method based on competency questions is presented by Uschold and King (1995).

A complete ontology development process, *METHONTOLOGY*, is proposed by Fernández et al. (1997). The process is composed by the following phases: specification, conceptualization, formalization, integration, implementation, maintenance. Its life cycle is based on evolving prototypes and specific techniques peculiar to each activity. Other activities, like control, quality assurance, knowledge acquisition, integration, evaluation and documentation are carried out simultaneously with the ontology development activities.

With a strong emphasis on knowledge maintenance and management, Sure et al. (2004) propose *On-To-Knowledge*, an ontology development process consisting of five main phases: feasibility study, kick-off, refinement, evaluation, application and evolution. Each phase consists of a number of sub-steps.

Other approaches, often tied to industry or research projects, include the methods used for building *CyC*, *SENSUS*, and *KAKTUS* (OntoWeb deliverable, 2002). A complete overview of ontology building methods is provided by Corcho et al. (2003).

We provide a two-fold evaluation of the proposed approach. First, we provide a comparative evaluation with respect to the methodologies introduced above. Second, we briefly describe our experience in using the process in building an ontology of *eProcurement* for the Athena Integrated Project.

In order to evaluate a number of different ontology building processes, Fernández and Gómez-Pérez (2002) present a framework based on the comparison with respect to the IEEE 1074-1995 standard for software development life cycle. Here we integrate UPON into the evaluation framework in order to assess it with respect to the other proposals.

The IEEE standard, applied to ontologies, distinguishes three kinds of processes: *project management processes*, concerning the creation of a project management frame-

work for the entire ontology life cycle; *ontology development processes*, including a *pre-development* process (an environment study and a feasibility study), a *development* process (requirements, design, implementation) and a *post-development* process (installation, operation, support, maintenance and retirement of an ontology); *integral processes*, required to complete ontology project activities.

Because of its nature, UPON does not deal with project management processes and pre/post development activities, while this is a major benefit of the On-To-Knowledge approach. On the other side, the adoption of UPON does not require any learning curve for domain experts using UML and the Unified Process, because it is just an adaptation of the UP to ontology building. This is an advantage also over the adoption of METHONTOLOGY, that roughly covers the same development processes as UPON. Furthermore, an extension of the UP, the *Enterprise Unified Process* (Nalbone et al., 2004), is being developed with the aim of taking project management and all the other pre/post development activities into account. In the future we will consider the extension of UPON to these other aspects.

Another big advantage of UPON over the other methodologies is that diagramming, documentation and versioning can be performed with the aid of a variety of tools specialized for UML, like Rational Rose, Microsoft Visio, etc.

UPON was applied in the context of the Athena Project for building an ontology of *eProcurement*. Despite its preliminary stage, both domain experts and modellers expressed their appreciation. The developed ontology consists of 23 actors, 21 processes, 14 objects and 83 attributes, complex and atomic. Though it may seem a "small ontology", it is appropriate for the given purposes. In particular it allows the semantic annotation of the main business documents (e.g. the purchase order and the invoice) used in a purchasing transaction.

4 Conclusions and Future Work

In this paper we presented UPON, an ontology building methodology based on the Unified Process. Ontology building is different from developing a software system, but we showed that the basic phases are the same and some diagrammatic specifications can be used for each phase of the lifecycle of both software systems and ontologies.

The strength of the approach lies in the UP being a highly scalable and customizable framework. It can indeed be tailored to fit a number of variables: the ontology size, the domain of interest, the complexity of the ontology to build, the experience and skill of the project organization and its people. Furthermore, the modellers can decide to adapt the scheme presented here for one of the methodologies derived from the UP (like the Rational Unified Process).

In a future work, we would like to provide a more detailed evaluation of the process with respect to the other proposals as well as an analysis of how to adapt cross-phase activities to the needs of ontology building. In describing UPON, some aspects of the UP, like interfaces, architectures, activity diagrams etc., have been neglected for the sake of space.

Finally, an important aspect is the possibility of assessing the quality of an ontology built with the UPON methodology. This issue is currently under elaboration and will be presented in the next future.

References

1. T. Berners-Lee, J. Hendler, O. Lassila (2001). The Semantic Web. *Scientific American*, may 2001.
2. O. Corcho, M. Fernández, A. Gómez-Pérez (2003). Methodologies, tools and languages for building ontologies. Where is the meeting point? *Data & Knowledge Engineering*, 46.
3. M. Fernández, A. Gómez-Pérez, N. Juristo. (1997) METHONTOLOGY: From Ontological Art towards Ontological Engineering. *Symposium on Ontological Engineering of AAAI*. Stanford, California.
4. M. Fernández, A. Gómez-Pérez (2002). Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, **17**(2).
5. A. Gómez-Pérez, M. Fernández-Lopez, O. Corcho (2004). *Ontological Engineering*, Springer-Verlag, London.
6. T. R. Gruber (1993). A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition* **5**, pp. 199-220.
7. M. Gruninger and M. S. Fox. Methodology for the Design and Evaluation of Ontologies, Proc. of *Workshop on Basic Ontological Issues in Knowledge Sharing* in IJCAI 95, Montreal, Canada, 1995.
8. N. Guarino, M. Carrara, P. Giaretta (1994). Formalizing Ontological Commitments. In *Proceedings of AAAI 94*, volume 1, pp. 560-567.
9. G. Guizzardi, H. Herre, G. Wagner (2002). Towards Ontological Foundations for UML Conceptual Models. *1st International Conference on Ontologies, Databases and Application of Semantics*, Irvine, California, USA.
10. I. Jacobson, G. Booch, and J. Rumbaugh (1999). *The Unified Software Development Process*. Addison Wesley, USA.
11. M. Missikoff, F. Taglino (2002). Business and Enterprise Management with SymOntoX. 1st International Semantic Web Conference, Sardinia, Italy.
12. R. Navigli, P. Velardi (2004). Learning Domain Ontologies from Document Warehouses and Dedicated Websites, *Computational Linguistics* 30(2), MIT Press, June.
13. J. Nalbone, M. Vizdos, M. Ambler. *Adopting the Enterprise Unified Process*. White paper, Ronin International Inc., 2004.
14. *OntoWeb Deliverable 1.4: A Survey on Methodologies for Developing, Maintaining, Evaluating and Reengineering Ontologies* (2002). http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D1.4-v1.0.pdf
15. J. Sowa (1999). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole, USA.
16. Y. Sure, S. Staab, R. Studer (2004). On-To-Knowledge Methodology (OTKM). *Handbook on Ontologies*, Springer, pp. 117-132.
17. M. Ushold, and M. Gruninger: Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2) (1996).
18. M. Uschold and M. King: Towards a Methodology for Building Ontologies. Proc. of Workshop on Basic Ontological Issues in Knowledge Sharing in IJCAI 1995, Montreal, Canada (1995).