

SEMANTIC TECHNOLOGY INSTITUTE (STI)



A COST MODEL FOR LIGHTWEIGHT
ONTOLOGIES: ADAPTING THE
ONTOCOM MODEL

Igor O. Popov

STI TECHNICAL REPORT 2008-08-29
AUGUST 2008

SEMANTIC TECHNOLOGY INSTITUTE (STI)

STI Innsbruck
University of Innsbruck
Technikerstrasse 21a
Innsbruck, Austria
www.sti-innsbruck.at



STI TECHNICAL REPORT

STI TECHNICAL REPORT 2008-08-29, AUGUST 2008

A COST MODEL FOR LIGHTWEIGHT ONTOLOGIES: ADAPTING THE ONTOCOM MODEL

Igor O. Popov¹

Abstract. In this report we investigate the possibility of adapting ONTOCOM, a cost estimation method for ontologies, for a class of lightweight ontology structures, like taxonomies. We examine both the ontology development process and the taxonomy development process in which we point out the similarities and contrasts and subsequently propose several adaptations. Similar to ONTOCOM, we propose that any adaptations are evaluated by using the quality framework proposed by [3]. Our report ends with a section on future work.

Keywords: semantic web, lightweight ontologies, cost estimation, taxonomy development.

¹Semantic Technology Institute (STI) Innsbruck, University of Innsbruck,
Technikerstraße 21a, A-6020 Innsbruck, Austria. E-mail: igor.popov@sti-innsbruck.at

Acknowledgements: This work was supported by the European Commission under the project ACTIVE.

Copyright © 2008 by the authors

Contents

1	Introduction and motivation	1
2	ONTOCOM Model	2
2.1	Cost Estimation methodologies	2
2.2	The ONTOCOM Model	3
2.2.1	Top-down breakdown	3
2.2.2	The ONTOCOM Cost Drivers	4
2.2.3	The Parametric Formula	5
2.3	Calibration of the ONTOCOM model	5
3	Simple vs. complex ontologies	6
4	Taxonomy development process	7
4.1	Requirement analysis	7
4.2	Identifying concepts	8
4.3	Developing a draft taxonomy	9
4.4	Review with Users and domain experts	10
4.5	Taxonomy refinement	10
4.6	Instantiation	10
4.7	Manage and maintain taxonomy	10
5	Cost Drivers for simple ontologies	10
5.1	Product Factors	10
5.1.1	Domain complexity: DCPLX	10
5.1.2	Concept derivation complexity: CDCPLX	12
5.1.3	Classification complexity: CCPLX	12
5.1.4	Taxonomy Evaluation: TE	12
5.1.5	Documentation needs: DOCU	13
5.1.6	Classification of Data: CDATA	13
5.1.7	Taxonomy Maintenance: TM	14
5.2	Personnel Factors	14
5.2.1	Taxonomy/Domain expert capability: TECAP/DCAP	14
5.2.2	Taxonomy/Domain expert experience: TXEXP/DEXP	14
5.2.3	Tool/Language experience: TEXP/LEXP	14
5.2.4	Personnel Continuity: PCON	15
5.3	Project Factors	15
5.3.1	Support tools for Taxonomy Development: TOOL	15
5.3.2	Multisite Development: SITE	16
5.3.3	Required Development Schedule: SCED	16
6	Evaluation	16
7	Conclusion and future work	16

1 Introduction and motivation

Over the past decade ontologies and their development have become increasingly popular and have become one of the well established research areas in the Semantic Web. Indeed, this increased popularity is evident by the surge of tools, methodologies and research associated with ontology development. Certain groups within the Semantic Web community are now committed to the task of studying the ontology development life cycle in order to provide a well-defined and structured engineering process for building ontologies. This discipline is now commonly referred to as Ontology Engineering [8, 9, 11, 18, 17]. Ontology Engineering, borrowing the ideas from the more mature discipline of Software Engineering, tries to identify common development patterns in ontology developments in the same way as those identified in the software development life cycle. Current research done on the subject suggests that the ontology development cycle is an iterative process that bears similarities with the software development process. As ontology engineering matures into a well-defined process which deals with real-life problems in businesses, government and education, the challenges it will have to address will go beyond the technical to include economic ones as well.

A first attempt made to investigate the economic aspects of ontology development has been made with the Ontology Cost Estimation Model (ONTOCOM) [13, 14, 15]. ONTOCOM has been dealing with estimating the development effort (in Person Months - PM) needed to build an ontology, taking into account all the phases in the ontology life-cycle. Assuming that ontology engineering can utilize results in effort estimation from adjacent and more mature engineering fields, like software engineering, ONTOCOM uses the well know parametric approach of the Constructive Cost Model (COCOMO) [4, 3, 6] to derive a similar cost model for ontologies. Like COCOMO, ONTOCOM applies a parametric formula to calculate the effort in person months, statistically calibrating the formula based on expert input and historical data from developers.

ONTOCOM has achieved promising results in the estimation of effort in ontology development. However, developed to be a generic model, ONTOCOM considers ontologies from the entire range of the ontology spectrum [12]: controlled vocabularies, taxonomies, taxonomy with properties, heavy weight ontologies etc. From the data gathered and developer interviews conducted, we have noticed that the type, environment and development mode all play a significant role in explaining the effort of building ontologies, and that many of the data points divided along those lines show different behavior w.r.t. to effort.

To investigate these effects and improve the precision of the ONTOCOM model we want to introduce variants of ONTOCOM which will reflect the particularities of the development of the individual structures (or a more common set of structures) in the ontology spectrum. Specifically, we would like to restrict ourselves to the more lightweight structures in the ontology spectrum since we believe that the original ONTOCOM model encompasses well the attributes of developing heavy-weight ontologies. We are further motivated the fact that lightweight ontologies are the most commonly occurring type of ontologies on the Semantic Web [7].

Since every cost estimation model should be based on some structured development process we believe that taxonomies are a good starting point where our efforts should start. Taxonomy development has certainly been in practice for a much longer time and is a more mature field within the area of knowledge and information management. Their use by also is common; most companies and institutions today has some sort of classification within their ranks, be it related to the classification of data, services or other subjects. Organizations, such as UNSPSC [1], and scientific communities like the medical community have already deployed standards using a taxonomy structure to classify its contents. Building taxonomies has thus become a well structured process in the area of knowledge management. The American National Standards

Institution (ANSI) has already published a guideline for the construction, format and management of controlled vocabularies [2]. Since taxonomies represent a simpler expression of knowledge than ontologies we believe that ONTOCOM could be adapted to specifically target taxonomies and related structures.

The remainder of this paper is organized as follows. The next section gives a brief overview of the ONTOCOM model, its cost drivers and calibration methods. In Section 3 we outline the class of taxonomy structures we take into consideration and we examine differences between simple and complex ontologies. In Section 4 we present the taxonomy development process and in Section 5 we propose a set of refined cost drivers suited to the taxonomy development process explained in Section 4. We end this paper with a section on conclusions and future work.

2 ONTOCOM Model

2.1 Cost Estimation methodologies

Cost estimation for engineering purposes can be approached in several ways. To select a suitable approach for developing a cost estimation model the general characteristics of the engineering task are examined which allow cost drivers (factors, which have an impact on the overall effort) to be deduced. To further improve it, a model for cost estimation can be tuned to reflect the specific setting of the environment w.r.t. the project, personnel and process aspects. In a real-world situation however, it is seldom the case that a single strategy will satisfy all aspects for a model so it is common that a model is based on several strategies.

The common strategies used and at the same time those considered by ONTOCOM include:

Analogy Method. The main idea of the analogy method states that the cost associated with similar projects should have similar costs.

Bottom-Up Method. This method tries to identify specific components and estimate the costs associated with the development of each component and subsequently calculate the overall effort as the sum of its parts.

Top-Down Method. This method is basically the opposite of the Bottom-Up approach, applicable in situations where at an early stage of the project the components cannot be identified and only global properties are known. It therefore offers a top-down partition where the partition is done at a certain phase in the project where such a partition is justifiable.

Expert Judgment/Delphi Method. This method involves a structured process of data collection based on expert opinion about the efforts associated with different aspects of the project.

Parametric/Algorithmic Method. This approach tries to use a mathematical formula to calculate the effort based on a statistical analysis of data from previous projects. This method uses the statistical analysis to improve accuracy and find dependencies between cost factors.

Again, assuming that engineering practices have similar properties, ONTOCOM adopts the top-down, parametric and expert based methods as viable methods to develop its cost estimation model. This is further supported by the fact that the work breakdown structure in ontology engineering is similar to that found in software engineering practices.

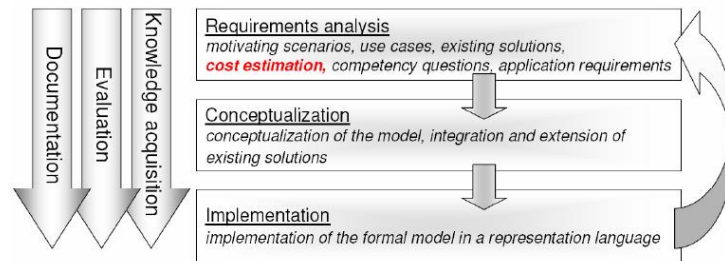


Figure 1: The ontology engineering process

2.2 The ONTOCOM Model

The ONTOCOM model is realized in three steps that reflect the strategies that were mentioned in the previous subsection. First, a top-down work breakdown is done along the phases of the ontology engineering process. Second, a set of cost drivers and values associated with pre-defined intervals are proposed and evaluated by experts in the field of ontology development. Third, an a-priori model is proposed based on a mathematical formula after which empirical (historical) data from previous ontology building projects are gathered and used in conjunction with the expert data to statistically calibrate the model and analyze dependencies between cost drivers. This calibration results in a better and a validated a-posteriori model.

2.2.1 Top-down breakdown

The top-down partitioning considered by ONTOCOM is based on a study of several ontology development methodologies [15, 10]. It can be concluded that the core development steps in an ontology development project include:

- 1) **Requirements analysis.** The requirements analysis consists of task such as analysis of project settings based on a pre-determine set of requirements, knowledge gathering activities and use or reuse of any information sources.
- 2) **Conceptualization.** The application domain is modeled in terms of ontological primitives like, concepts, relation properties.
- 3) **Implementation.** The conceptual model is implemented in a language, whose expressivness is appropriate to the richness of the model.
- 4) **Evaluation.** The resulting ontology is evaluated in a manual, semi-automatic or automatic way after which the ontology can under go changes based on the results of the evaluation.

Other additional steps might evolve feasibility studies, ontology population (where the data is mapped to the ontology) and ontology maintenance.

A typical ontology engineering process is depicted in the Figure 1.

After a clear picture of the tasks in each step of the ontology development process cost drivers can be proposed.

2.2.2 The ONTOCOM Cost Drivers

Based on extensive literary surveys and interviews with experts on ontology engineering, ONTOCOM identifies factors, called Cost Drivers (CDs) which influence the overall impact on effort. These are grouped into three groups:

1) Product Factors which account for the influence of product properties on the overall costs.

Cost drivers for ontology building

Complexity of the Domain Analysis (DCPLX)

Complexity of the Conceptualization (CCPLX)

Complexity of the Implementation (ICPLX)

Complexity of the Instantiation (DATA)

Required Reusability (REUSE)

Documentation Needs (DOCU)

Complexity of the Ontology Integration (OI)

Complexity of the Ontology Evaluation (OE)

Cost drivers for reuse and maintenance

Complexity of the Ontology Evaluation (OE)

Complexity of the Ontology Modifications (OM)

Ontology Translation (OT)

Ontology Understanding (OU)

Ontologist / Domain Expert Unfamiliarity (UNFM)

2) Personnel Factors which consider factors associated with the team building the ontology, such as personnel capability and experience.

Ontologist / Domain Expert Capability (OCAP/DECAP)

Ontologist / Domain Expert Experience (OEXP / DEEXP)

Personnel Continuity (PCON)

Language and Tool Experience (LEXP / TEXP)

3) Project Factors which look at the environment settings which supports or hinders progress in the engineering process.

Tool Support (TOOL)

Multi-site Development (SITE)

Required Development Schedule (SCED)

Each cost driver is assigned five ratings (from Very Low to Very High) which should reflect a degree to which a cost driver had an impact on the development effort in a single ontology. For example, a High or Very High rating for the Domain Complexity (DCPLX) means that the domain modeled to ontology had a complex domain and that this had an high or very high impact on the development effort. Conversely, if the

domain modeled by the ontology is simple in nature the DCPLX rating for that ontology should be Low or Very Low. For the a-priori model each of these ratings corresponds to a numeric value i.e. a weight which is derived based on interviews with experts and is calculated as an average of their proposed values.

2.2.3 The Parametric Formula

ONTOCOM uses the following equation to calculate the necessary person-months:

$$PM = A * Size^\alpha * \prod CD_i \quad (1)$$

where Size is the size of the ontology, calculated in number of kilo entities (the sum of all concepts, properties, axiom and fixed instances), α is an exponential factor to account for any non-linear behavior in effort w.r.t. the size, A is a baseline multiplicative constant in person months, and CD_i are the weights of each cost driver commensurate to the rating assigned for each cost driver based on their role in the ontology development.

2.3 Calibration of the ONTOCOM model

Similar to other parametric models, ONTOCOM too relies on statistics based on previous project data to calibrate the model and thus create an a-posteriori model which will produce better estimates. ONTOCOM follows the calibration techniques described in [3], which try to refine the values (weights) on the ratings of the cost drivers by statistically tuning the values to reflect both the input from the experts and those of the historical data. What follows is a brief discussion of the calibration methods used by ONTOCOM.

The most common method to refine the values in existing cost estimation models is the classical multiple regression approach. The use of multiple regression on ONTOCOM first requires that the parametric formula is linearized which is done by taking logarithms on both sides of the equation:

$$\ln(PM) = \ln(A) + \alpha \ln(Size) + \sum \ln(CD_i) \quad (2)$$

A correlation analysis is then performed on the historical data to see where predictor variables have high correlation which may result in the aggregation of two or more cost drivers into a single cost driver.

While multiple regression is a popular method, it is not uncommon to have some counter intuitive results in the refined values e.g. having cost driver with ratings with High impact on effort actually having a values which cause a decrease in effort [6]. More on why such effects might occur can be found in [5].

To resolve such effects, ONTOCOM uses a Bayesian calibration approach [6] which is based on Bayesian Analysis - a well-defined and rigorous process of inductive reasoning. In short, the Bayesian calibration approach tries to unify the information from the historical data and the expert judgment by reasoning (or rather calculating) where it is more appropriate to "lean more" towards the expert opinion at the expense on the historical data and vice versa.

Using Bayesian analysis ONTOCOM calculates the scaling factors β^{**} for the cost drivers:

$$\beta^{**} = \left[\frac{1}{s^2} X'X + H^* \right]^{-1} \times \left[\frac{1}{s^2} X'X\beta + H^* \beta^* \right] \quad (3)$$

$$Var(\beta^{**}) = \left[\frac{1}{s^2} X'X + H^* \right]^{-1} \quad (4)$$

where X is the matrix of collected data, s^2 is the variance of the residual of the linear regression, and H^* is the inverse of the matrix containing the variance of the expert estimations and β^* is the matrix containing the means of expert estimations. A more detailed explanation of the theory on Bayesian analysis is given in [6].

Rating	Rating Scale
Very Low	concept list
Low	taxonomy, high number of patterns, no constraints
Nominal	properties, general pattern available, some constraints
High	axioms, few modeling pattern, considerable number of constraints
Very High	instances, no patterns, considerable number of constraints

Table 1: Ratings for the Complexity of Conceptualization.

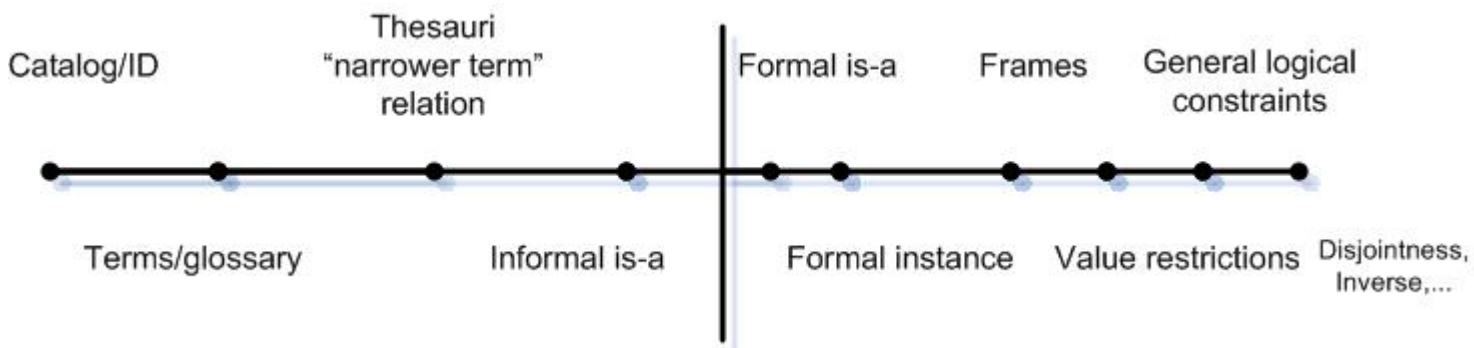


Figure 2: The ontology engineering process

3 Simple vs. complex ontologies

As we have seen ONTOCOM model tries to approximate the effort associated with building ontologies. ONTOCOM, however, considers all possible notions of ontologies in its model, and uses a conceptualization complexity factor to account for the impact the type of ontology has on the costs. The range of the ratings for the Complexity of Conceptualization factor (CCPLX) is presented in Table 1. As we stated earlier our goal is to adapt ONTOCOM for a subset of similar structures in the ontology spectrum. We assume that structures which have similar complexity and which are closely related in the manner in which they are developed will show more common behavior w.r.t to effort.

Defining an ontology (with specific emphasis on web ontologies) as a specification of the conceptualization of a domain, [12] divides the potential interpretations in a simple linear spectrum according to the detail in their specification (Figure 2). The first item in the spectrum is a simple controlled vocabulary - i.e. a finite list of terms. A catalog is an type of a controlled vocabulary which allows for unambiguous interpretation of terms by associating them with a unique identifier.

A glossary extends catalogs by providing unstructured meaning to the terms. These are usually specified in as natural language statements. A glossary is intended for human use and lacks any structure making them inadequate for machine processing.

Thesauri add semantics in the relationship between terms. It usually restricts these relationships to dealing with ambiguity and synonymy. Typically, thesauri do not provide explicit hierarchical relationships.

Taxonomies typically deal with defining a hierarchical relationship between concepts. [12] splits the ontology spectrum into structures which may be defined as ontologies based on the strictness of the "is-a" relationship in a taxonomy. A strict "is-a" relationship requires that if A is a superclass of B, then any object

of B is also an instance of A. Not all taxonomies however are strict taxonomies e.g. the classification scheme of Yahoo.

The formal "is-a" relationship introduces us into the realm of ontologies. Every following item on the ontology spectrum represents the expressiveness used to model the ontology. These include the introduction of frames, value restrictions, axioms and logical constraints.

McGuinness considers simple ontologies to hold the following properties:

- Finite controlled (extensible) vocabulary
- Unambiguous interpretation of classes and term relationships
- Strict hierarchical subclass relationships between classes

Properties which are considered typical but not mandatory are:

- Property specification on a per-class basis
- Individual inclusion in the ontology
- Value restriction specification on a per-class basis

The following properties are considered desirable, but neither mandatory, nor typical:

- Specification of disjoint classes
- Specification of arbitrary logical relationships between terms
- Distinguished relationships such as inverse and part-whole

For our purposes we consider both simple ontologies as defined above as well as informal taxonomy structures and control vocabularies. We believe that the development process associated with both kind of structures have similarities in their development process. This leads us to believe that the estimation of effort for both kinds of structures will exhibit similar properties as well. To adapt ONTOCOM to suit effort estimation for taxonomies and similar structures we will have to review the taxonomy development process and investigate which factors in a taxonomy development process may have an impact on effort.

4 Taxonomy development process

Similar to software engineering and ontology engineering, taxonomies are built in an iterative way. [19] defines a seven step iterative process shown in Figure 3. We briefly discuss what kind of issues each of the stages deals with. This will provide us with a overview of the tasks at each stage, their complexity and possible impact on effort.

4.1 Requirement analysis

Similar to the engineering processes previously mentioned, developing a taxonomy begins with determining the requirements. In taxonomy development, this starts with defining the scope, purpose, and types of content formats. In addition, identifying a target audience and interviews regarding the content of the taxonomy is done in this phase as well.

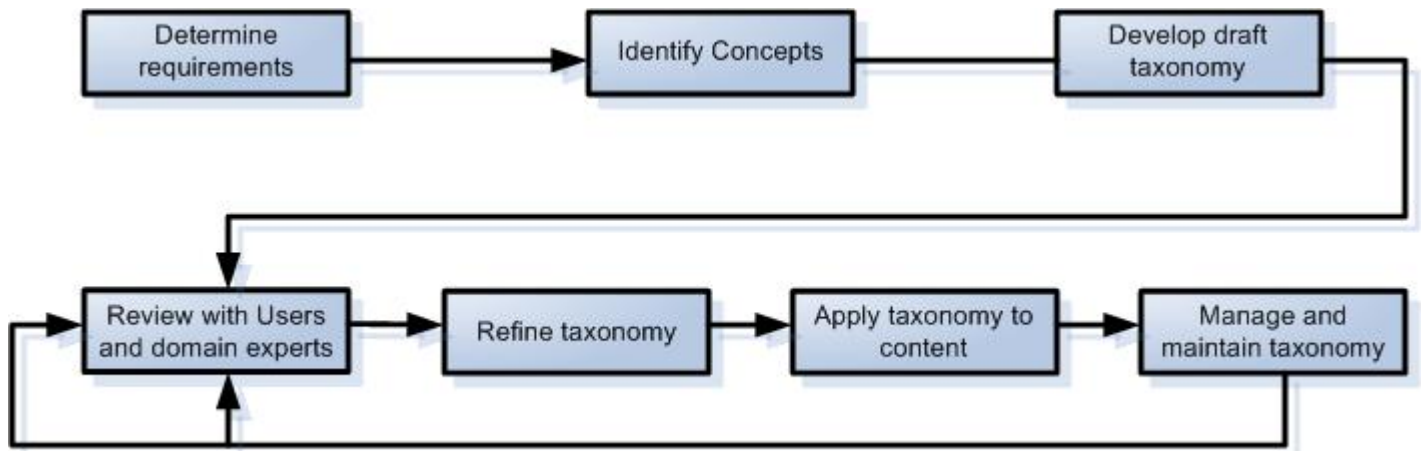


Figure 3: The ontology engineering process

The requirements should cover a variety of issues, such as what business objective does the taxonomy meet, why is the taxonomy used and what is it trying to solve. Then a survey with the domain experts and stakeholders should be performed to analyze the concepts which are important to them, whether they possess existing information sources for categorizing information etc. The third issue the requirements should resolve is any technological constraints that might have an impact on the taxonomy development.

4.2 Identifying concepts

Identifying concepts to be used in the taxonomy is the next step in the taxonomy development process. However, it is tied closely with the next step, creating the draft taxonomy, which is why identifying concepts and their relationships with one another is sometimes intertwined [2]. Identifying concepts can be a crucial phase w.r.t. effort since it evolves a comprehensive analysis of information sources, users and domain experts. The strategy of identifying concepts, the number of information sources and the complexity of the domain could all have a significant impact.

[2] identifies three approaches in identifying candidate concepts:

1) Committee approach. This approach basically puts the identification of concepts and their relationships and subsequent arrangement in the same phase. Experts in domain draw up a list of key concepts along with their relationships with assistance from the experts in taxonomy design. The list of concepts could have been derived from various sources and submissions from users or communities. The strategy of creating the taxonomy with the committee approach could have one of two possible choices:

- a. Top Down - First the broadest concepts are identified then narrower concepts are selected in subsequent steps to reach the level of specification desired. The hierarchical structure and any specific properties of the content are created as the work proceeds.
- b. Bottom Up - This approach is usually taken when a list of concepts have been identified from a corpus of content objects and then incorporated into a taxonomy. As in the top down approach,

the hierarchical structure and any of the properties are created as the work proceeds but now going from narrower concepts to more generic ones.

2) The Empirical Approach. The committee approach unified the concepts identification step and the taxonomy development step. In contrast, the Empirical approach, has two basic methods:

- a. The deductive method - First a method of collection of concepts are derived from various information sources and users without any attempt to set any hierarchical relationships between them. Collecting new concepts last until a sufficient number of concepts have been gathered after which a review is conducted together with domain experts and taxonomy design experts to judge the appropriateness of the selected concepts. Afterward, determining their class relationships is made in the next phase on a broad to narrower basis.
- b. The inductive method - In this method, new candidate concepts are immediately judged for potential inclusion in the taxonomy as they are encountered. This means that vocabulary control is applied from the outset of the project. The hierarchical structure is done at an ad hoc basis which each candidate concept designated whether it is a subclass of a broader class. The hierarchical structure is done at a narrower to broad basis.

3) Combination of Methods. In practice a combination of the first two methods is employed in the development process. For example, concepts and the hierarchy derived by using first the inductive method could later be reviewed with the deductive method. The methods and the order in which they are employed usually depends on the knowledge and experience at each stage.

The above methods assume that the decisions are made by humans. Machine assistance at a certain level could be helpful for identifying concepts and may reduce the effort associated with this stage. There are three areas in which machine assistance might be beneficial:

1. Identification of candidate concepts - Candidate concepts can be identified automatically from electronic information sources e.g. titles and abstracts from texts. Removing words such as conjunctions and prepositions could also be helpful in deriving candidate terms.
2. Registering frequency of concept assignment - A very high or a very low frequency could be considered for deletion or modification.
3. Recording concepts from user queries - Concepts found in user queries which are not included in the candidate concepts list may be considered for inclusion. They can also be assigned a weight based on the frequency of the concept in multiple user queries.

At this stage also a generic taxonomy can also be considered to be used as an input in the taxonomy. However, this taxonomy may need to be modified to satisfy the requirements, which might again have an impact on the effort. In addition taxonomies that capture similar domains may also be extremely useful in selecting candidate concepts.

4.3 Developing a draft taxonomy

As we mentioned earlier, the actual conceptualization of the taxonomy is done during as well as after the identification of the concepts. From a technical point of view this stage could also include some sort of technical implementation of the taxonomy in some language, metadata structure or other electronic formats. The technical implementation should also deal with other issues such as the extensibility of the taxonomy, flexibility, and reuse.

4.4 Review with Users and domain experts

Reviewing the initial and subsequent versions of the taxonomy might have a significant impact on the effort. This might be attributed to several factors such as high requirements, unpredicted changes in the model, amount of interaction between the personnel and personnel capability and experience. The reviewing process might include a variety of stakeholders, from domain experts to users and taxonomy experts. The process might include interviews, usability studies and technical testing of the taxonomy. It is therefore understandable that the evaluation of the taxonomy, similar to the evaluation of ontologies, will also play a factor in estimating effort.

4.5 Taxonomy refinement

Based on the feedback from the users and domain experts, the taxonomy is refined to incorporate agreed-to changes. In this stage the taxonomy usually undergoes several reviews and refinements.

4.6 Instantiation

Applying a taxonomy to the content usually means using some sort of application which will use the taxonomy for services such as search, browsing and navigation or be used as a control vocabulary. Instantiation can be a major factor in the overall effort since mapping resources to the taxonomy can be done in a manual, semi-automatic or automatic way.

4.7 Manage and maintain taxonomy

Most taxonomies are subject to periodic reviewing and changes. These reviews are can be related to both structure and content of the taxonomy. At the highest level these changes could be due a change in general functionality requirements or purpose of the taxonomy. More low-level changes might include reviewing, testing or adding new content. Moreover when the taxonomy is integrated into other systems, it is vital to have a process for managing version control. All these aspects point out that maintenance could have an impact on the effort.

5 Cost Drivers for simple ontologies

Similar to the original ONTOCOM cost model the adaptation of ONTOCOM for lightweight structures will differentiate between product, project and personnel cost drivers. Project and personnel cost drivers relate to similar aspects in every engineering process, thus they need only to be slightly adapted to the engineering process underlying the cost model. Most of the changes are in the product cost drivers where the task is to identify new cost drivers based on the complexity of the product, refine similar cost drivers which reflect the characteristics of the product, and remove any cost drivers which might not apply to taxonomy development.

5.1 Product Factors

5.1.1 Domain complexity: DCPLX

The domain complexity driver examines the complexity of the domain from which the taxonomy is built, the number and complexity of the requirements and the availability of information sources. As is the case

with ontologies, the domain in which the taxonomy is built can be wide or narrow and might require expert knowledge or it can be a common-sense knowledge area. The requirements in taxonomy building can cover several aspects, such as design and technical aspects which should cover the issues related with determining the scope, the purpose and content object of the taxonomy, and user related requirements. We also account for the impact the availability of information sources can bring to the taxonomy development process. The rating scales for the domain complexity, requirements and information sources are given in Table 2,3,4:

Rating	Rating Scale
Very High	wide scope, expert knowledge, high connectivity
High	moderate to wide scope, common-sense or expert knowledge, high connectivity
Nominal	moderate to wide scope, common-sense or expert knowledge, moderate connectivity
Low	narrow to moderate scope, common-sense or expert knowledge, low connectivity
Very Low	narrow scope, common-sense knowledge, low connectivity

Table 2: Domain complexity.

Rating	Rating Scale
Very High	very high number of req. with a high conflicting degree, high number of usability requirements
High	high number of usability requirements, few conflicting requirements
Nominal	moderate number of requirements, with few conflicts, few usability requirements
Low	small number of non-conflicting requirements
Very Low	few simple requirements

Table 3: Requirements complexity

Rating	Rating Scale
Very High	high number of information sources and structured data, high use of a generic taxonomy
High	high number of information sources, some modifications to a generic taxonomy
Nominal	good quality and number of information sources
Low	some information sources availability
Very Low	none

Table 4: Information Sources.

5.1.2 Concept derivation complexity: CDCPLX

As we saw in the previous section one of the biggest challenges in taxonomy development is to derive the concepts for the taxonomy. The Concept derivation complexity cost driver CDCPLX account for the impact problems such as synonyms and ambiguity cases have on the process of deriving the concepts. In addition it also factors in any machine assistance associated with this stage of the taxonomy development.

Rating	Rating Scale
Very High	Mostly manual work, very high number of synonyms, ambiguity cases
High	High amount of manual work, some automatic processing, high level of synonyms, ambiguity cases
Nominal	Manual work done in combination with some automatic processing, some instances of synonym and ambiguity cases
Low	Some manual work, high use of automatic processing, rare instances of synonym and ambiguity cases
Very Low	A few manual steps mostly reviewing, high use of automatic processing, nearly no instances of synonym and ambiguity cases

Table 5: Concept derivation complexity.

5.1.3 Classification complexity: CCPLX

The Classification Complexity (CCPLX) cost driver measures the effort associated with establishing the hierarchical relationships between the concepts. The classification complexity is a separate cost driver from the Concept Derivation Complexity (CDCPLX) cost driver, so our model could include simpler structures that do not have hierarchical relationships such as lists. In such a case this cost driver should be discarded.

Rating	Rating Scale
Very High	Difficulty in establishing hierarchical relationships for almost every concept, high number of multi-inheritance relationships
High	High difficulty in establishing hierarchical relationships, some number of multi-inheritance relationships
Nominal	Moderate difficulty in establishing hierarchical relationships
Low	Some cases of difficulty in establishing hierarchical relationships
Very Low	Nearly no difficulty in establishing hierarchical relationships

Table 6: Classification complexity.

5.1.4 Taxonomy Evaluation: TE

Taxonomy evaluation can undergo extensive reviewing with domain experts and user as well as testing. [16] gives some of the criteria for evaluating a taxonomy. To determine the effort associated with evaluation, the TE cost drivers tries to scale the scope and intensity of the evaluation process.

Rating	Rating Scale
Very High	Extensive reviews and testing
High	Considerable number of reviews and testing
Nominal	Moderate number of reviews and testing
Low	Small and low number of reviews and tests
Very Low	Almost no reviews and tests

Table 7: Taxonomy Evaluation.

5.1.5 Documentation needs: DOCU

Similar to ontology development and all other complex engineering processes, additional costs may arise as a consequence of documentation requirements during the lifecycle process of development.

Rating	Rating Scale
Very High	comprehensive documentation for every stage in the LC process
High	comprehensive documentation only for some stages in the LC process
Nominal	right-sized documentation for every stage in the LC process
Low	some stages omitted from documentation needs
Very Low	many stages omitted from documentation needs

Table 8: Documentation needs.

5.1.6 Classification of Data: CDATA

Classification of data in a taxonomy involves mapping the content to the concepts of the taxonomy. This means that the content somehow has to be mapped using either some language, schema or database. The effort associated with this cost driver corresponds to the degree of automation that is possible.

Rating	Rating Scale
Very High	unstructured data in natural language, free form, manual mapping required
High	semi-structured data in natural language, e. g. similar web pages, some automation available
Nominal	semi-structured, automation available however needs manual reviewing
Low	structured data, high degree of automation, manual intervention minimal
Very Low	structured data, process completely automated

Table 9: Classification of Data.

5.1.7 Taxonomy Maintenance: TM

The Taxonomy Maintenance cost driver TM tries to determine the impact on effort on of the modifications during reviews and the number of new concepts and data added.

Rating	Rating Scale
Very High	Substantial reorganization of the hierarchy and reclassification of data
High	Significant reorganization of the hierarchy and reclassification of data
Nominal	Some reorganization of the hierarchy and reclassification of data
Low	Minor reorganization of the hierarchy and reclassification of data
Very Low	No reorganization of the hierarchy and reclassification of data

Table 10: Taxonomy Maintenance.

5.2 Personnel Factors

5.2.1 Taxonomy/Domain expert capability: TECAP/DCAP

Similar to the ontology development process, the taxonomy development process will also require the collaboration of personnel with background in building taxonomies and domain experts and stakeholders which posses the knowledge about the domain. Like ONTOCOM the cost drivers account for the perceived ability of the actors involved in the process as well as their performance as a team.

	Very Low	Low	Nominal	High	Very High
TECAP/DCAP	15%	35%	55%	75%	90%

Table 11: Capability Ratings of the Engineering Team.

5.2.2 Taxonomy/Domain expert experience: TXEXP/DEXP

These factors try to measure the experience of the two teams as a whole in their respective areas, taxonomy development and domain conceptualization.

Like ONTOCOM, this cost driver tries to evaluate the experience of the taxonomy developers with the tools at their disposal and/or language e.g. markup languages like XML or languages like OWL. This also includes knowledge of any representation languages used during the processes of identifying and classifying concepts.

5.2.3 Tool/Language experience: TEXP/LEXP

Like ONTOCOM, this cost driver tries to evaluate the experience of the taxonomy developers with the tools at their disposal and/or language e.g. markup languages like XML or languages like OWL. This also includes knowledge of any representation languages used during the processes of identifying and classifying concepts.

	Very Low	Low	Nominal	High	Very High
TXEXP	6 months	1 year	3 years	5 years	7 years
DEXP	6 months	1 year	3 years	5 years	7 years

Table 12: Experience Ratings for the Team

	Very Low	Low	Nominal	High	Very High
TEXP	2 months	6 months	1 year	2 years	3 years
DEXP	2 months	6 months	1 year	2 years	3 years

Table 13: Tool and Language Experience

5.2.4 Personnel Continuity: PCON

The personnel continuity cost driver tries to factor in the changes in personnel during the process life-cycle in a time and resource constrained project. Since we believe that the size of the engineering team in ontology development and taxonomy development are likely the same we use the same values proposed in the original ONTOCOM model:

	Very Low	Low	Nominal	High	Very High
TECAP/DCAP	50%	35%	25%	15%	10%

Table 14: Personnel Continuity

5.3 Project Factors

5.3.1 Support tools for Taxonomy Development: TOOL

Support tools for taxonomy development during all stages of the development life-cycle surely have a great impact on effort. While we expect that most taxonomy development team have a variety of tools at their disposal we would like to account for their effectiveness in all the stages of development. A set of tools can include tools for design, implementation and mapping of data to the taxonomy. It can also include tools which help in concept identification as those described in Section 4.2.

Rating	Rating Scale
Very High	High quality and availability of tools, manual intervention minimal
High	Few manual processing required
Nominal	Basic manual intervention needed
Low	Some tool support
Very Low	Minimal tool support, mostly manual processing

Table 15: Tool Support

5.3.2 Multisite Development: SITE

Like in ontology development, taxonomy development might require extensive communication between the various parties. The SITE cost driver assesses the communication support tools:

Rating	Rating Scale
Very High	frequent F2F meetings
High	teleconference, occasional meetings
Nominal	email
Low	phone, fax
Very Low	mail

Table 16: Multisite Development

5.3.3 Required Development Schedule: SCED

The SCED cost driver accounts for the impact of schedule constraints on the taxonomy development process. Processes which have tight schedule (under 100%) constraints tend to produce more effort in the later stages of the taxonomy development lifecycle like refinement and evolution. Stretched-out schedule usually produce more effort early on like in the identification of concept stage.

	Very Low	Low	Nominal	High	Very High
TECAP/DCAP	75%	85%	100%	130%	160%

Table 17: Required Development Schedule

6 Evaluation

Like ONTOCOM, the proposed cost drivers for the modification of ONTOCOM for lightweight structures will rely on the quality framework proposed by Boehm. The framework consists of 10 evaluation criteria assuring quality in a variety of aspects. A brief description of them is provided in Table 15 (for more see [3]): Similarly, the process will have to be conducted in two steps: first the a-priori model and the cost drivers are evaluated by experts, second the a-posteriori model is evaluated by observing the quality of the predictions. Being at an early stage, this report tries to identify an initial set of cost drivers which can be submitted for evaluation by experts. After this task is completed, gathering data points and calibration of the model for an a-posteriori validation could be considered.

7 Conclusion and future work

The aim of this report was to attempt to adapt the cost estimation model for ontologies, ONTOCOM, for lightweight ontologies like taxonomies. Such an adaptation is intended to potentially improve the precision of ONTOCOM. The rationale behind such an adaptation is that similar structures would share same complexity and similar development steps which would lead to similar costs.

No	Criterion	Description
1	Definition	-clear definition of the estimated and the excluded costs -clear definition of the decision criteria used to specify the cost drivers -intuitive and non-ambiguous terms to denominate the cost drivers
2	Objectivity	- objectivity of the cost drivers and their decision criteria
3	Constructiveness	- human understandability of the model predictions
4	Detail	- accurate phase and activity breakdowns
5	Scope	- usability for a wide class of ontology engineering processes
6	Ease of use	- easily understandable inputs and options - easily assessable cost driver ratings based on the decision criteria
7	Prospectiveness	- model applicability in early phases of the project
8	Stability	- small differences in inputs produce small differences in outputs
9	Parsimony	- lack of highly redundant cost drivers - lack of cost drivers with no appreciable contribution to the results
10	Fidelity	- reliability of the predictions

Table 18: Evaluation Framework

Like ONTOCOM, this model has defined cost drivers by examining the cost related to three factors: product, personnel and project. Since most of the factors in engineering practices associated with personnel and the project environment are similar, as expected, most of the adaptation done to ONTOCOM was in relation to the product factors. In particular we tried to map the peculiarities of the taxonomy development process to define new, more concrete, complexity cost drivers and modify the existing ones.

Subsequent work will have to investigate in detail the practice of reuse in taxonomy development. In particular a clearer distinction should be made between a reuse scenario as those define in ontology development and those used in taxonomy development which include using generic taxonomic structures or using existing taxonomy structures importing them directly into taxonomy as opposed to using them as information sources. Additionally factors such as maintenance and evolution of taxonomies should also be investigated. At present we have a preliminary model which will have to undergo an evaluation from experts and in the future be calibrated based on real world data.

Acknowledgments This work was supported by the European Commission under the project Active.

References

- [1] Unspsc homepage. <http://www.unspsc.org/>.
- [2] Guidelines for the construction, format, and management of monolingual controlled vocabularies. Technical Report ANSI/NISO Z39.19-2005, National Information Standards Organization, 2005.
- [3] B. W. Boehm. *Software Engineering Economics*. Prentice-Hall, 1981.
- [4] B. W. Boehm, C. Abts, B. Clark, and S. Devnani-Chulani. Cocomo ii model definition manual. 1997.
- [5] B. C. Briand and W. M. Basili, V. R. and Thomas. A pattern recognition approach for software engineering data analysis. *IEEE Transactions on Software Engineering*, 18(11), November 1992.

- [6] S. Chulani. Incorporating bayesian analysis to improve the accuracy of cocomo ii and its quality model extension. phd thesis. 1998.
- [7] J. Davis, D. Fensel, and F. van Harmelen, editors. *Towards the Semantic Web: Ontology-driven Knowledge Management*. Wiley, 2002.
- [8] M. Fernandez, A. Gomez-Perez, and N. Juristo. Methontology: From ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium on Ontological Engineering*, 1997.
- [9] M. Fernandez-Lopez and A. Gomez-Perez. Overview and analysis of methodologies for building ontologies. *Knowledge Engineering Review*, (17(2):129156), 2002.
- [10] A. Gomez-Perez, M. Fernandez, and O. Corcho. *Ontological Engineering with examples from areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer, 2004.
- [11] M. Gruninger and M. Fox. Methodology for the design and evaluation of ontologies. In *Proceedings of the IJCAI95, Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
- [12] D. L. McGuinness. Ontologies come of age. In D. Fensel, J. Hendler, H. Lieberman, and C. Wahlster, editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.
- [13] E. Paslaru Bontas and M. Mochol. "a cost model for ontology engineering". Technical Report TR-B-05-03, Freie Universitt Berlin, April 2003.
- [14] E. Paslaru Bontas Simperl and C. Tempich. Towards a cost estimation model for ontology engineering. In *Proceedings of the 3rd Berliner XML Tage*, Berlin, Germany, 2005.
- [15] E. Paslaru Bontas Simperl, C. Tempich, and Y. Sure. Ontocom: A cost estimation model for ontology engineering. In *Proceedings of the International Semantic Web Conference (ISWC) 2006*, Atlanta, USA, 2006.
- [16] Soergel and Dagobert. Thesauri and ontologies in digital libraries: Tutorial. In *Evaluation of thesauri. Joint Conference on Digital Libraries*, 2002.
- [17] Y. Sure, S. Staab, and R. Studer. On-to-knowledge methodology. In *Handbook on Ontologies*, pages 117–132. 2004.
- [18] M. Uschold and M. King. Methodology for the design and evaluation of ontologies. In *Proceedings of the IJCAI95, Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
- [19] M. Whittaker and K. Breininger. Taxonomy development for knowledge management. proceedings of the world library and information congress. 2008.