Towards a Cost Estimation Model for Ontology Engineering

Elena Paslaru Bontas, Malgorzata Mochol
Freie Universität Berlin
Institut für Informatik
Takustr. 9 D-14195 Berlin, Germany
paslaru@inf.fu-berlin.de, mochol@inf.fu-berlin.de

Abstract: This paper introduces ONTOCOM, a parametric cost estimation model for Semantic Web ontologies. After analyzing established, general-purpose cost estimation methodologies we propose a methodology, which can be applied to develop cost models for ontology engineering. We examine the particularities of this engineering field on the basis of the proposed methodology, in order to identify cost factors which influence the effort invested in ontology building, reuse and maintenance.

1 Introduction

Though ontologies and associated ontology management tools have become increasingly popular in the last decades, a wide-scale dissemination of ontologies and ontology-based applications as envisioned by the Semantic Web community requires fine-grained methodologies which address both *technical* and *economical* challenges of ontology engineering. In order for ontologies to be built and deployed at a large scale, beyond the boundaries of the academic community, one needs not only technologies to assist the development process, but also proved and tested means to control economical aspects such as the associated costs. A wide range of ontology engineering methodologies have emerged in the Semantic Web community[FLGP02]. Apart from minor differences in the level of detail adopted for the description of the process stages these methodologies define ontology engineering as an iterative process, which shows obvious similarities to the neighbored research field of software engineering. Cost estimation is addressed only marginally by current engineering approaches (e.g. in [LTGP04]), though its importance is well recognized in the community. In this paper we present a *methodology* for the development of cost models for ontologies and analyze *cost factors* implied in the engineering process. Likewise software engineering we describe a parametric approach to ontology cost estimation and propose the prototypical cost model **ONTOCOM** (**ONTO**logy **CO**st **M**odel).[1]

2 A Methodology for Cost Estimation

Cost Estimation is defined as an art of predetermining the lowest realistic cost/price of an item or activity which assure a normal profit. In the case of Ontology Engineering a cost

[1]This work is a result of the cooperation within the Semantic Web PhD-Network Berlin-Brandenburg and has been partially supported by KnowledgeWeb - Network of Excellence and by the projects"A Semantic Web for Pathology" funded by the DFG (German Research Foundation) and "Knowledge Nets", which is part of the InterVal- Berlin Research Centre for the Internet Economy, funded by the German Ministry of Research BMBF.

model aims at *predicting the costs* (efforts in person months or duration) related to activities performed during the life cycle of an ontology. Estimating costs can be performed according to several methodologies [St95, Bo81], which, due to their limitations w.r.t. certain classes of situations, are usually applied in conjunction for the improvement of the predicted results. We examined the suitability of some of the most important ones w.r.t. ontology engineering, given the *current* state of the art in the Semantic Web area:

Analogy method The main idea of this methodology is the extrapolation of available data from similar projects to estimate the costs of the proposed project. In terms of ontologies, this method would require a reliable comparison function for ontologies as well as cost information from previous ontology engineering projects. While several similarity measures for ontologies have already been proposed in the Semantic Web community (e.g. in [MS02]), no case studies on ontology costs are currently available.

Bottom-Up method This approach involves identifying and estimating costs of individual project components separately and subsequently combining the outcomes to produce an estimation for the overall project. It can not be applied early in the life cycle of the process because of the lack of necessary information. In our case the bottom-up method is currently not feasible, since it assumes the availability of cost information w.r.t. single engineering tasks (e.g. costs of the domain conceptualization).

Top-Down method The counterpart top-down approach relies on global parameters of the proposed project. For this purpose, the project is partitioned into lower-level components and life cycle phases beginning at the highest level. This method is more applicable to early cost estimates when only global properties are known. Despite the young nature of the ontology engineering discipline, the Semantic Web community has made significant progresses in analyzing the particularities of ontology engineering process and several fine-grained methodologies have already been applied to various application domains with very promising results. Therefore a top-down approach can be applied to define a framework in which additional cost estimation methodologies can be employed to predict costs associated with single stages of the engineering process (see below).

Expert judgment/Delphi method This method is based on a structured process for collecting and distilling knowledge from a group of human experts (using their past project experiences) by means of a series of questionnaires interspersed with controlled opinion feedback. The expert judgement method seems to be appropriate for our goals since large amount of expert knowledge w.r.t. ontologies is already available in the Semantic Web community, while the costs of the related engineering efforts are not. Experts' opinion on this topic can be used to compliment the results of other estimation methods.

Parametric/Algorithmic method This method uses mathematical equations derived from research and historical data from previous projects to compute the costs arisen in a specific project. It involves the identification of cost drivers for a specific class of projects and uses statistical techniques to customize the corresponding equations. The results of the method are directly related to the availability of reliable and relevant data to be used in calibrating the core model. Though such information is not available for our purposes yet, the parametric approach can be applied to define a preliminary non-calibrated cost model (the so-called "a-priori cost model") which is subject to constant refinements (resulting in

the so-called "a-posteriori cost model"). The analysis of the main cost drivers affecting the ontology engineering process – an important step towards the elaboration of a predictable cost estimation strategy – can be performed on the basis of existing case studies on ontology building, maintenance and reuse. Besides, the definition of a fixed spectrum of cost factors is fundamental for a controlled data collection.

To summarize, *top-down, parametric and expert-based approaches* can be partially used to develop a cost estimation methodology for ontologies. [2] Due to the lack of complete and reliable cost-related information, a combination of the three methodologies is likely to perform better than single approaches. Figure 1 illustrates the life cycles of the methodology we applied to generate a cost estimation model for ontology engineering. We started with a top level approach, by identifying upper-level sub-tasks of a standard ontology engineering process[FLGP02], defined the associated costs using the parametric and the expert-based methods[PBM05]. Expert judgement is used to evaluate a preliminary set of cost drivers and to specify their start values in the a-priori model (see below). We distinguished among

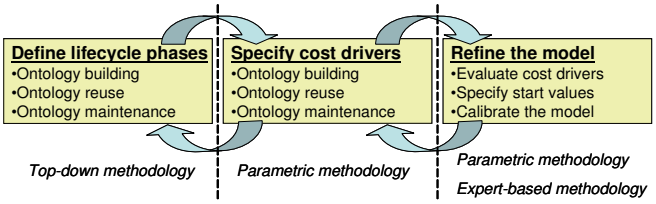

Figure 1: Cost Model Development Methodology

three areas, whose costs are to be defined separately:[3]

Ontology Building includes the typical stages of an ontology engineering process[FLGP02]: domain analysis (result: requirements specification), the conceptualization (result: conceptual model), the implementation (result: specification of the conceptual model in the selected representation language) and the ontology population i.e. the generation of instances and their alignment to the model (result: instantiated ontology).[4]

Ontology Maintenance includes getting familiar with and modifying the ontology (insert or delete new ontological primitives, re-model parts of the ontology etc.)

Ontology Reuse involves costs for the discovery and re-usage of existing (source) ontologies in order to generate a new (target) ontology. The latter includes understanding, evaluating and adapting the former ones to the requirements of the latter.

[2] A bottom-up or an analogy approach will be relevant solely after defining and testing a higher-level cost estimation model and collecting large amounts of cost information from real-world projects.

[3] In order to increase the usability of the model w.r.t. a wide range of ontology engineering methodologies, we plan to refine the aforementioned process stages and to align the high-level cost drivers to these extensions.

[4] At this point we consider solely manual ontology building. Automatic ontology building methods such as ontology learning from texts are out of the scope of this paper.

3 ONTOCOM: The ONTOlogy COst Model

ONTOCOM is a parametric cost estimation model for ontologies which aims at predicting the effort invested in building, maintaining and reusing ontologies on the basis of predefined cost drivers. Starting from a typical ontology engineering scenario, in which an ontology is developed – from scratch, by adapting existing knowledge sources, or both – and deployed/maintained by its users, ONTOCOM calculates the necessary effort (expressed in person months) using the following equation:

$$PM = PM_B + PM_M + PM_R \quad (1)$$

PM_B , PM_M and PM_R represent the person months associated to building, maintaining and reusing ontologies, respectively and are calculated as:

$$PM_X = Size_X * \prod CD_{Xi} \quad (2)$$

Each of the three development phases is associated with *specific* cost factors. Experiences in related engineering areas[Ke87, Bo81] let us assume that the most significant one is the *size of the ontology* involved in the corresponding process. In the formula above the size parameter $Size_X$ is expressed in thousands of ontological primitives – concept, relations, axioms and instances. [5] $Size_B$ corresponds to the size of the newly built ontology i.e. the number of primitives which are expected to result from the conceptualization phase. In case of ontology maintenance the size of the ontology ($Size_M$) depends on the expected number of modified items. For reuse purposes the relevant factor $Size_R$ is the size of the original source after being tailored to the present application setting. In particular this involves the parts of the source ontologies which have to be translated to the final representation language, the ones whose content has to be adapted to the target scope and the fragments directly integrated. The *cost drivers* CD_{Xi} have a rating level (from very low to very high) that expresses their impact on the development effort. For the purpose of a quantitative analysis, each rating level of each cost driver is associated to a weight (effort multiplier - EM). The average EM assigned to a cost driver is 1.0 (nominal weight). If a rating level causes more development effort, its corresponding EM is above 1.0. If the rating level reduces the effort then the corresponding EM is less than the nominal value. In the a-priori cost model a team of 3 ontology engineering experts assigned start values between 0.1 and 2 to the effort multipliers, depending on the contribution of the corresponding cost driver to the overall development costs.[6] These values are subject of further calibration on the basis of the statistical analysis of real-world project data. In the following we turn to a brief description of the cost drivers in ONTOCOM.[7] These parameters were derived after surveying recent literature and from empirical findings of various case studies in the ontology engineering field (such as [PBMT05, UHW+98, RVMS99]). For each cost driver we specified in detail the decision criteria which are relevant when assigning the corresponding effort multipliers. For example for the cost driver LEXP – accounting for costs produced by the level of experience of the engineering team w.r.t. ontology representation languages – we pre-defined the meaning of the effort multipliers as

[5] For example for an ontology with 1000 concepts and 100 relations $Size$ will have the value 1.1.

[6] A list of these values is available in [PBM05].

[7] See [PBM05] for a detailed explanation of the approach.

depicted in Table 1. The values of the corresponding effort multipliers, which have been specified by human experts, are as follows: 1.30 (Very Low), 1.15 (Low), 1 (Nominal), 0.85 (High) and 0.75 (Very High)[PBM05]. The suitable value is selected during the cost estimation procedure and is used as a multiplier in equation 2.

	Very Low	**Low**	**Nominal**	**High**	**Very High**
LEXP	2 months	6 months	1 year	3 years	6 years

Table 1: Language Experience LEXP

3.1 Cost Drivers for Ontology Building

For the ontology building area we defined a list of cost drivers, which are, similar to [Bo97], divided into three groups:

Product-related cost drivers account for the influence of ontology characteristics on the overall costs: i) Instance (DATA) to capture the effects that the instance data requirements have on the overall process, ii) Ontology Complexity (OCPLX) to express those ontology features which increase the complexity of the engineering outcomes, iii) Required Reusability (REUSE) to capture the additional effort associated with the development of a reusable ontology, and iv) Documentation match to life-cycle needs (DOCU) to state for the additional costs caused by very detailed documentation requirements.

Project-related cost drivers relate the dimensions of the engineering process and its characteristics to the overall costs: i) Support tools for Ontology Engineering (TOOL) to measure the effects of using ontology management tools in the engineering process, ii) Multisite Development (SITE) to mirror the usage of the communication support tools in a location-distributed team, and iii) Required Development Schedule (SCED) to measure the effects certain schedule constraints have on the development effort.

Personnel-related cost drivers emphasize the role of team experience, ability and continuity w.r.t. the effort invested in the process: i) Ontologist/Domain Expert Capability (OCAP/DECAP) to account the perceived ability and efficiency of the single actors involved in the process (ontologist and domain expert) as well as their teamwork capabilities, ii) Ontologist/Domain Expert Experience (OEXP/DEEXP) to measure the level of experience of the engineering team w.r.t. performing ontology engineering activities, iii) Language/Tool Experience (LEXP/TEXP) to measure the level experience of the project team w.r.t. the representation language and the ontology management tools, and iv) Personnel Continuity (PCON) to mirror the frequency of the changes in the project team.

3.2 Cost Drivers for Ontology Reuse and Maintenance

Additionally to project and personnel cost drivers (as described in Section 3.1) we defined a set of further 4 cost drivers to deal with the characteristics of ontology reuse and maintenance, as reported by relevant case studies in these areas[PBMT05, UHW$^+$98, RVMS99]:

Ontology Understanding(OU) accounts for the efforts required to get familiar with the ontologies to be used, a task which is a pre-condition to ontology evaluation and maintenance. It depends on ontology properties such as representation language or size and on

the level of experience of the ontology engineer w.r.t. this ontology[PBM05].

Ontology Evaluation(OE) accounts for the additional efforts related to the evaluation phase given a satisfactory ontology understanding level (e.g. for testing the source ontologies against a specific set of requirements or for documenting the approach).

Ontology Modification/Translation(OM/OT) are factors reflecting the costs involved by adapting the source ontologies to the new setting (e.g. inserting, deleting ontology concepts) and by translating to a target representation language, respectively.

4 Related Work

As mentioned in the previous sections estimating costs is a fundamental requirement for a wide-scale dissemination of ontologies in business contexts. However, though the importance of cost issues is well-recognized in the community, no cost estimation model for ontology engineering is available so far. Cost estimation methods have a long-standing tradition in more mature engineering disciplines such as software engineering or in industrial production. Approaches in these areas[Bo81, Ke87, St95] offered us valuable information about methods which can be applied to define and evaluate ONTOCOM.

5 Future Work

ONTOCOM is an a-priori cost estimation model for the ontology engineering area. This model is intended to predict cost arisen during ontology engineering processes by analyzing the costs factors caused by the end product, the engineering process and the involved personnel. Starting from an analysis of general-purpose estimation methodologies we proposed a methodology to deduce ontology costs and specified costs factors implied by ontology building, reuse and maintenance. The presented cost model is to be continuously improved by engineering experts and refined with the collection of empiric data on person month efforts invested in developing ontologies in real-world projects and on the basis of the Delphi method. Besides, we are performing a detailed quality assessment of the mentioned cost drivers on the basis of the framework by Boehm[Bo81].

References

[Bo81] Boehm, B. W.: *Software Engineering Economics*. Prentice-Hall. 1981.

[Bo97] Boehm, B. W. and Abts, C. and Clark, B. and Devnani-Chulani, S. COCOMO II Model Definition Manual. 1997.

[FLGP02] Fernández-López, M. and Gómez-Pérez, A.: Overview and analysis of methodologies for building ontologies. *Knowledge Engineering Review*. 17(2):129–156. 2002.

[Ke87] Kemerer, C. F.: An Empirical Validation of Software Cost Estimation Models. *C-ACM*. 30(5). 1987.

[LTGP04] Lozano-Tello, A. and Gomez-Perez, A.: ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management, Vol. 15, No. 2*. 15(2). 2004.

[MS02] Maedche, A. and Staab, S.: Measuring Similarity between Ontologies. In: *Proc. of the European Conf. on Knowledge Acquisition and Management EKAW-2002*. 2002.

[PBM05] Paslaru Bontas, E. and Mochol, M.: A cost model for ontology engineering. Technical Report TR-B-05-03. FU Berlin. April 2005.

[PBMT05] Paslaru Bontas, E., Mochol, M., and Tolksdorf, R.: Case Studies in Ontology Reuse. In: *Proc. of the 5th Int. Conf. on Knowledge Management IKNOW05*. 2005.

[RVMS99] Russ, T., Valente, A., MacGregor, R., and Swartout, W.: Practical Experiences in Trading Off Ontology Usability and Reusability. In: *Proc. of the Knowledge Acquisition Workshop KAW99*. 1999.

[St95] Stewart, R. D. and Wyskida, R. M. and Johannes, J. D.: *Cost Estimator's Reference Manual*. Wiley. 1995.

[UHW+98] Uschold, M., Healy, M., Williamson, K., Clark, P., and Woods, S.: Ontology Reuse and Application. In: *Proc. of the Int. Conf. on Formal Ontology and Information Systems FOIS98*. S. 179–192. 1998.