ONTOCOM: A Cost Estimation Model for Ontology Engineering

Elena Paslaru Bontas Simperl[1], Christoph Tempich[2], and York Sure[3]

[1]Free University of Berlin, Takustr. 9, 14195 Berlin, Germany
paslaru@inf.fu-berlin.de
[2,3]Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
[2]tempich,[3]sure@aifb.uni-karlsruhe.de

Abstract. The technical challenges associated with the development and deployment of ontologies have been subject to a considerable number of research initiatives since the beginning of the nineties. The economical aspects of these processes are, however, still poorly exploited, impeding the dissemination of ontology-driven technologies beyond the boundaries of the academic community. This paper aims at contributing to the alleviation of this situation by proposing ONTOCOM (Ontology Cost Model), a model to predict the costs arising in ontology engineering processes. We introduce a methodology to generate a cost model adapted to a particular ontology development strategy, and an inventory of cost drivers which influence the amount of effort invested in activities performed during an ontology life cycle. We further present the results of the model validation procedure, which covered an expert-driven evaluation and a statistical calibration on 36 data points collected from real-world projects. The validation revealed that ontology engineering processes have a high learning rate, indicating that the building of very large ontologies is feasible from an economic point of view. Moreover, the complexity of ontology evaluation, domain analysis and conceptualization activities proved to have a major impact on the final ontology engineering process duration.

1 Introduction

The popularity of ontologies grows with the emergence of the Semantic Web. Nevertheless, their large scale dissemination – in particular beyond the boundaries of the academic community – is inconceivable in the absence of methods which address the *economic* challenges of ontology engineering processes in addition to the *technical* and *organizational* ones. A wide range of ontology engineering methodologies have been elaborated in the Semantic Web community [6]. They define ontology development as a well-structured process, which shows major similarities with established models from the neighboring area of software engineering. Unlike adjacent engineering disciplines these methodologies, however, ignore the economic aspects of engineering processes, which are fundamental in real-world business contexts. Topics such as costs estimation, quality assurance procedures, process maturity models, or means to monitor the business value and the impact of semantic technologies at corporate level have been marginally exploited so far.

This paper aims at contributing to the alleviation of this situation. We introduce ONTOCOM (Ontology Cost Model), a model for predicting the costs related to ontology engineering processes. In this context we describe a *methodology* to generate a cost model suitable for particular ontology development strategies, and an inventory of *cost drivers* for which we demonstrate to have a direct impact on the amount of effort invested during an ontology life cycle. ONTOCOM has been subject to an extensive validation procedure. This covered two phases: an expert-driven evaluation and a statistical calibration, which adjusted the predictions of the model according to 36 data points collected from empirical ontology engineering processes.

The remaining of this paper is organized as follows: Section 2 examines general-purpose cost estimation methods w.r.t. their relevance for the ontology engineering field. Building upon the results of this analysis Section 3 gives a detailed description of the ONTOCOM cost prediction model and explains how it can be applied to arbitrary ontology engineering processes. Section 4 discusses the results of the evaluation. We conclude the paper with related and future work (Section 5).

2 Cost Estimation Methodologies

In order to reliably approximate the development efforts the engineering team needs to specify a method for cost estimation in accordance with the particularities of the current project as regarding product, personnel and process aspects. This specification task can be accomplished either by building a new cost model with the help of dedicated methodologies or by adapting existing general-purpose ones to the characteristics of a specific setting.

Due to its high relevance in real-world situations cost estimation is approached by a wide range of methods, often used in conjunction in business context due to their optimal applicability to particular classes of situations. We give an overview of some of the most important ones [1, 10, 15]:

1) Analogy Method. The main idea of this method is the extrapolation of available data from similar projects to estimate the costs of the proposed project. The method is suitable in situations where empirical data from previous projects is available and trustworthy. It highly depends on the accuracy in establishing real differences between completed and current projects.

2) Bottom-Up Method. This method 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 related to the project components. Nevertheless since the costs to be estimated are related to more manageable work units, the method is likely to produce more accurate results than the other approaches.

3) Top-Down Method. This method relies on overall project parameters. For this purpose the project is partitioned top-down into lower-level components and life cycle phases (so-called *work breakdown structures* [1, 10]). The method is applicable to early cost estimates when only global properties are known, but it can be less accurate due to the decreased focus on lower-level parameters and technical challenges. These are usually predictable later in the process life cycle, at most.

4) Expert Judgment/Delphi Method. This approach is based on a structured process for collecting and distilling knowledge from a group of human experts by means of a series of questionnaires interspersed with controlled opinion feedback. The involvement of human experts using their past project experiences is a significant advantage of this approach. The most extensive critique point is related to the subjectivity of the estimations and the difficulties to explicitly state the decision criteria used by the contributors.
5) Parametric/Algorithmic Method. This method involves the usage of mathematical equations based on research and previous project data. The method analyzes main cost drivers of a specific class of projects and their dependencies, and uses statistical techniques to adjust the corresponding formulas. The generation of a proved and tested cost model using the parametric method is directly related to the availability of reliable project data to be used in calibrating the model.

Given the current state of the art in ontology engineering the **top-down**, **parametric** and **expert-based** methods form a viable basis for the development of a cost estimation model in this field.[1] A combination of the three is considered in many established engineering disciplines as a feasible means to reach a balance between the low amount of reliable historical data and the accuracy of the cost estimations [1, 15]. The work breakdown structure for ontology engineering is to a great extent described by existing ontology engineering methodologies. Further on, the cost drivers associated with the parametric method can be derived from the high number of case studies available in the literature. The limited amount of accurate empirical data can be counterbalanced by taking into account the significant body of expert knowledge available in the Semantic Web community. The next section describes how the three methods were jointly applied to create ONTOCOM.

3 The ONTOCOM Model

The cost estimation model is realized in three steps. First a *top-down* work breakdown structure for ontology engineering processes is defined in order to reduce the complexity of project budgetary planning and controlling operations down to more manageable units [1, 10]. The associated costs are then elaborated using the *parametric* method. The result of the second step is a statistical prediction model (i.e. a parameterized mathematical formula). Its parameters are given start values in pre-defined intervals, but need to be calibrated on the basis of previous project data. This empirical information complemented by expert estimations is used to evaluate and revise the predictions of the initial *a-priori model*, thus creating a validated *a-posteriori model*.

3.1 The Work Breakdown Structure

The top-level partitioning of a generic ontology engineering process can be realized by taking into account available process-driven methodologies in this field.[2] According to them ontology building consists of the following core steps (cf. Figure 1):

[1] By contrast the bottom-up method can not be applied in early stages of the ontology engineering process, while the analogy method requires means to compare among ontologies and associated development processes.

[2] Refer, for instance, to [6] for a recent overview on ontology engineering methodologies.

1) Requirements Analysis. The engineering team consisting of domain experts and ontology engineers performs a deep analysis of the project setting w.r.t. a set of predefined requirements. This step might also include **knowledge acquisition** activities in terms of the re-usage of existing ontological sources or by extracting domain information from text corpora, databases etc. If such techniques are being used to aid the engineering process, the resulting ontologies are to be subsequently customized to the application setting in the conceptualization/implementation phases. The result of this step is an ontology requirements specification document [16]. In particular this contains a set of competency questions describing the domain to be modelled by the prospected ontology, as well as information about its use cases, the expected size, the information sources used, the process participants and the engineering methodology.
2) Conceptualization. The application domain is modelled in terms of ontological primitives, e. g. concepts, relations, axioms.[3]
3) Implementation. The conceptual model is implemented in a (formal) representation language, whose expressivity is appropriate for the richness of the conceptualization. If required reused ontologies and those generated from other information sources are translated to the target representation language and integrated to the final context.
4) Evaluation. The ontology is evaluated against the set of competency questions. The evaluation may be performed automatically, if the competency questions are represented formally, or semi-automatically, using specific heuristics or human judgement. The result of the evaluation is reflected in a set of modifications/refinements at the requirements, conceptualization or implementation level.

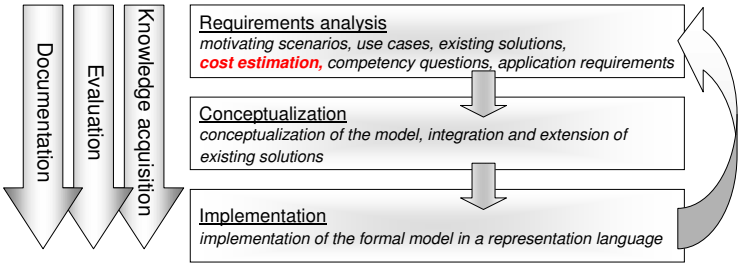

Fig. 1. Typical Ontology Engineering Process

Depending on the ontology life cycle underlying the process-driven methodology, the aforementioned four steps are to be seen as a sequential workflow or as parallel activities. Methontology [6], which applies prototypical engineering principles, considers **knowledge acquisition**, **evaluation** and **documentation** as being complementary *support activities* performed in parallel to the main development process. Other methodologies, usually following a classical waterfall model, consider these support activities as part of a sequential engineering process. The OTK-Methodology [16] additionally introduces an initial **feasibility study** in order to assess the risks associated with an ontology building attempt. Other optional steps are **ontology population/instantiation** and **ontology evolution/maintenance**. The former deals with the alignment of concrete

[3] Depending on methodology and representation language these ontological primitives might have different names, e.g. class or concept, relation or relationship, slot, axiom, constraint.

application data to the implemented ontology. The latter relates to modifications of the ontology performed according to new user requirements, updates of the reused sources or changes in the modelled domain. Further on, likewise related engineering disciplines, reusing existing knowledge sources—in particular ontologies—is a central topic of ontology development. In terms of the process model introduced above, **ontology reuse** is considered a **knowledge acquisition** task.

The parametric method integrates the efforts associated with each component of this work breakdown structure to a mathematical formula as described below.

3.2 The Parametric Equation

ONTOCOM calculates the necessary person-months effort using the following equation:

$$PM = A * Size^{\alpha} * \prod CD_i \tag{1}$$

According to the parametric method the total development efforts are associated with cost drivers specific for the ontology engineering process and its main activities. Experiences in related engineering areas [1, 7] let us assume that the most significant factor is the *size of the ontology* (in kilo entities) involved in the corresponding process or process phase. In Equation 1 the parameter $Size$ corresponds to the size of the ontology i.e. the number of primitives which are expected to result from the conceptualization phase (including fragments built by reuse or other knowledge acquisition methods). The possibility of a non-linear behavior of the model w.r.t. the size of the ontology is covered by parameter α. The constant A represents a baseline multiplicative calibration constant in person months, i.e. costs which occur "if everything is normal". The *cost drivers* CD_i 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_i). The *productivity range* PR_i of a cost driver (i.e. the ratio between the highest and the lowest effort multiplier of a cost driver $PR_i = \frac{max(EM_i)}{min(EM_i)}$) is an indicator for the relative importance of a cost driver for the effort estimation [1]. In the a-priori cost model a team of five ontology engineering experts assigned productivity ranges between 1.75 and 9 to the effort multipliers, depending on the perceived contribution of the corresponding cost driver to the overall development costs. The final effort multipliers assigned to the rating levels are calculated such that the contribution of an individual rating level is linear and the resulting productivity range for a cost driver corresponds to the average calculated from the expert judgements. In the same manner, the start value of the A parameter was set to 3.12. These values were subject to further calibration on the basis of the statistical analysis of real-world project data (cf. Section 4).

3.3 The ONTOCOM Cost Drivers

The ONTOCOM cost drivers, which are expected to have a direct impact on the total development efforts, can be roughly divided into three categories:
1) PRODUCT-RELATED COST DRIVERS account for the impact of the characteristics of the product to be engineered (i.e. the ontology) on the overall costs. The following cost

drivers were identified for the task of ontology building:
• **Domain Analysis Complexity (DCPLX)** to account for those features of the application setting which influence the complexity of the engineering outcomes,
• **Conceptualization Complexity (CCPLX)** to account for the impact of a complex conceptual model on the overall costs,
• **Implementation Complexity (ICPLX)** to take into consideration the additional efforts arisen from the usage of a specific implementation language,
• **Instantiation Complexity (DATA)** to capture the effects that the instance data requirements have on the overall process,
• **Required Reusability (REUSE)** to capture the additional effort associated with the development of a reusable ontology,
• **Evaluation Complexity (OE)** to account for the additional efforts eventually invested in generating test cases and evaluating test results, and
• **Documentation Needs (DOCU)** to state for the additional costs caused by high documentation requirements.
2) PERSONNEL-RELATED COST DRIVERS emphasize the role of team experience, ability and continuity w.r.t. the effort invested in the engineering process:
• **Ontologist/Domain Expert Capability (OCAP/DECAP)** to account for the perceived ability and efficiency of the single actors involved in the process (ontologist and domain expert) as well as their teamwork capabilities,
• **Ontologist/Domain Expert Experience (OEXP/DEEXP)** to measure the level of experience of the engineering team w.r.t. performing ontology engineering activities,
• **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,
• **Personnel Continuity (PCON)** to mirror the frequency of the personnel changes in the team.
3) PROJECT-RELATED COST DRIVERS relate to overall characteristics of an ontology engineering process and their impact on the total costs:
• **Support tools for Ontology Engineering (TOOL)** to measure the effects of using ontology management tools in the engineering process, and
• **Multisite Development (SITE)** to mirror the usage of the communication support tools in a location-distributed team.

The ONTOCOM cost drivers were defined after extensively surveying recent ontology engineering literature and conducting expert interviews, and from empirical findings of numerous case studies in the field.[4] For each cost driver we specified in detail the decision criteria which are relevant for the model user in order for him to determine the concrete rating of the driver in a particular situation. For example for the cost driver CCPLX—accounting for costs produced by a particularly complex conceptualization—we pre-defined the meaning of the rating levels as depicted in Table 1. The human experts assigned in average a productivity range of 6.17 to this cost driver. The resulting non-calibrated values of the corresponding effort multipliers are as follows: 0.28 (Very Low), 0.64 (Low), 1 (Nominal), 1.36 (High) and 1.72 (Very High) [11]. The appropriate value should be selected during the cost estimation procedure and used as a multiplier in

[4] See [11, 12] for a detailed explanation of the approach.

equation 1. Depending on their impact on the overall development effort, if a particular activity increases the nominal efforts, then it would be rated with values such as High and Very High. Otherwise, if it causes a decrease of the nominal costs, then it would be rated with values such as Low and Very Low.

Rating Level	Effort multiplier	Description
Very Low	0.28	concept list
Low	0.64	taxonomy, high nr. of patterns, no constraints
Nominal	1.0	properties, general patterns available, some constraints
High	1.36	axioms, few modelling patterns, considerable nr. of constraints
Very High	1.72	instances, no patterns, considerable nr. of constraints

Table 1. The Conceptualization Complexity Cost Driver **CCPLX**

The decision criteria associated with a cost driver are typically more complex than in the previous example and might be sub-divided into further sub-categories, whose impact is aggregated to the final effort multiplier of the corresponding cost driver by means of normalized weights [11, 12].

3.4 Using ONTOCOM in Ontology Engineering Processes

ONTOCOM is intended to be applied in early stages of an ontology engineering process. In accordance to the process model introduced above the prediction of the arising costs can be performed during the feasibility study or, more reliably, during the requirements analysis. Many of the input parameters required to exercise the cost estimation are expected to be accurately approximated during this phase: the expected size of the ontology, the engineering team, the tools to be used, the implementation language etc.[5]

The high-level work breakdown structure foreseen by ONTOCOM can be further refined depending of the ontology development strategy applied in an organization in a certain application scenario. As explained in Section 3.1 ONTOCOM distinguishes solely between the most important phases of ontology building: requirements analysis, conceptualization, implementation, population, evaluation and documentation. Further on, it focuses on *sequential* development processes (as opposed to, for instance, rapid prototyping, or iterations of the building workflow). In case the model is applied to a different ontology development process, the relevant cost drivers are to be aligned (or even re-defined) to the new sub-phases and activities, while the parametric equation needs to be adapted to the new activity breakdown. An example of how ONTOCOM can be applied to an ontology development methodology targeted at rapid prototyping in distributed scenarios is provided in [12].

After this optional customization step the model can be utilized for cost predictions.[6] For this purpose the engineering team needs to specify the rating levels associated with each cost driver. This task is accomplished with the help of decision criteria

[5] Ontology engineering methodologies foresee this information to be collected in a ontology requirements document at the end of this phase [16].

[6] However, if new cost drivers have been defined in addition to the ones foreseen by ONTOCOM, these should be calibrated using empirical data.

which have been elaborated for each of the cost driver rating levels (such as those for the CCPLX cost driver illustrated in Figure 2). Cost drivers which are not relevant for a particular scenario should be rated with the nominal value 1, which does not influence the result of the prediction equation.

4 Evaluation

For the evaluation of the model we relied on the quality framework for cost models by Boehm[1], which was adapted to the particularities of ontology engineering. The framework consists of 10 evaluation criteria covering a wide range of quality aspects, from the reliability of the predictions to the model ease-of-use and its relevance for arbitrary ontology engineering scenarios (Table 2).

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 2. The ONTOCOM Evaluation Framework

The evaluation was conducted in two steps. First a team of experts in ontology engineering evaluated the a-priori model, in particular the ONTOCOM cost drivers, w.r.t. their relevance to cost issues (Criteria 1 to 8 in the table above) . Second the predictions of the model were compared with 36 observations from real world projects (Criteria 9 and 10 of the quality framework).

4.1 The Expert-based Evaluation

The evaluation of the a-priori model was performed by conducting interviews with two groups of independent experts in the area of ontology engineering. Considering that the people best placed to give a comprehensive assessment of the cost estimation model are IT practitioners or researchers being directly involved in theoretical or practical issues of ontology engineering, we organized two experts groups affiliated in both communities, which evaluated the model sequentially. The first group consisted of 4 academics whose research was in the area of Semantic Web and Ontology Engineering. The second group brought together 4 researchers and 4 IT senior managers from companies with a Semantic Web profile. Participants were given a one hour overview of the ONTOCOM approach, followed by individual interviews. We summarize the key findings

of the conducted interviews categorized according to the criteria depicted in Table 2:
• **Definition/Constructiveness** The first draft of the model did not include the ontology evaluation activity. The cost driver **Evaluation Complexity (OE)** was introduced to the model for this purpose. The **Ontology Instantiation (OI)** cost driver was extended with new decision criteria and minor modifications of the terminology were performed.
• **Objectivity** The objectivity of the cost drivers and the associated decision criteria were evaluated by the participants favorably. Both suffered minor modifications. W.r.t. the size of the ontology, a key parameter of the model, some of the participants expressed the need for a more careful distinction between the impact of the different types of ontological primitives (e.g. concepts, axioms, relationships) w.r.t. the total efforts. In particular, as axioms and relationships between concepts are more challenging to be modelled than simple concepts and taxonomical structures, they recommended that this difference should be reflected by the parametric model. While the current version of ONTOCOM does not include this option, we are investigating the possibility of introducing a revised size formula which associates particular ontology primitives' categories to normalized weights:

$$Size = w_1 * NoClasses^{\alpha_1} + w_2 * NoRelations^{\alpha_2} + (1 - w_1 - w_2) * NoAxioms^{\alpha_3} \quad (2)$$

A final direction w.r.t this issue is planed for the a-posteriori model, as we require a significant set of empirical data in order to prove the validity of the experts' recommendations.
• **Detail/Scope** The cost drivers covered by the model were unanimously estimated to be relevant for the ontology engineering area. The collection of empirical data demonstrated that the model accommodates well to many real-world settings, situation which was also confirmed by applying ONTOCOM to the DILIGENT ontology engineering methodology[12]. However, the majority of the evaluators emphasized the need of a revised model for reuse and evolution purposes, an issue which will be investigated in the future. W.r.t. the detail of the cost drivers covered by the model, three new product drivers stating for the complexity of the domain analysis, conceptualization and implementation (DCPLX, CCPLX and ICPLX, see Section 3.3) were introduced in return to an original cost driver **Ontology Complexity** (OCPLX). Some of the participants also expressed the need for a more detailed coverage of the ontology evaluation task in engineering processes, so as to distinguish between the evaluation of an ontology against a set of competency questions and its fitness of use within a particular software system. A final decision w.r.t. this modification requires, however, a more significant set of empirical data.
• **Ease of use** The goal and the scope of the model were easily understood by the interviewees. During the data collection procedure, the only factor which seemed to require additional clarification was the size of the ontology, which was conceived to cover all types of ontological primitives (e.g. concepts/classes, properties, axioms, rules, constraints, manually built instances). Further on, the experiments revealed that there is no clear understanding between the re-usage of existing ontologies and the acquisition of ontologies from more un-structured knowledge sources such as text documents. However, this latter issue can not be necessarily considered as a weakness of the model itself,

but as the result of a potentially ambiguous definition of the two activities in current ontology engineering methodologies.
• **Prospectiveness** Some of the participants manifested concerns w.r.t. the availability of particular model parameters in early phases of the engineering process. However, as underlined in a previous section, many of the input parameters are foreseen to be specified in the ontology requirements specification document in the last part of the requirements analysis phase.
• **Stability** This is ensured by the mathematical model underlying ONTOCOM.

4.2 Evaluation of the Prediction Quality

The remaining two evaluation criteria **Fidelity** and **Parsimony** were approached after the statistical calibration of the model. In order to determine the effort multipliers associated with the rating levels and to select non-redundant cost drivers we followed a three-stage approach: First experts estimated the a-priori effort multipliers based on their experience as regarding ontology engineering. Second we applied linear regression to real world project data to obtain a second estimation of the effort multipliers.[7] Third we combined the expert estimations and the results of the linear regression in a statistically sound way using Bayesian analysis [2].

Data Collection The results reported in this paper are based on 36 structured interviews with ontology engineering experts [13]. The interviews were conducted within a three months period and covered 35 pre-defined questions related to the aforementioned cost drivers. The survey participants are representative for the community of users and developers of semantic technologies. The group consisted of individuals affiliated to industry or academia, who were involved in the last 3 to 4 years in ontology building projects in areas such as skill management, human resources, medical information systems, legal information systems, multimedia, Web services, and digital libraries.[8] The average number of ontology entities in the surveyed ontologies is 830 with a median at 330. It took the engineers in average 5.3 month (median 2.5) to build the ontologies. 40% of the ontologies were built from scratch. Reused ontologies contributed in average 50% (median 50%) of ontology entities to the remaining 60% of the surveyed ontologies.

Data Analysis In order to adapt the prediction model in accordance to experiences from previous ontology engineering processes we derived estimates of the cost driver productivity ranges from the collected data set. The estimates were calculated following a linear regression approach combined with Bayesian analysis. This approach allows the usage of human judgement and data-driven estimations in a statistically consistent way, such that the variance observed in either of the two determines its impact to the final values.[9] Linear regression models perform better with an increasing number of incor-

[7] Linear regression is a mathematical method to calculate the parameters of a linear equation so that the squared differences between the predictions from the linear equation and the observations are minimal [14].

[8] Around 50% of the interviewees were affiliated to industry.

[9] Refer to [4] for an exhaustive explanation of the application of Bayesian analysis for cost estimation purposes.

porated observations and a decreasing number of parameters to estimate. Its drawbacks can be compensated with the help of human estimations [4] and by excluding those parameters which have an insignificant influence on the final prediction value or are highly correlated.

In order to select relevant cost drivers for the ONTOCOM model we performed a correlation analysis on the historical data (Table 3). We excluded the following cost

Cost driver	**Correlation with PM**	**Cost driver**	**Correlation with PM**	**Comment**
SIZE	0.50	DATA	0.31	strong correlation with DCPLX
OE	0.44	SITE	0.27	low number of different data points
DCPLX	0.39	DOCU	0.22	moderated influence; strong
REUSE	0.38			correlation with OE
ICPLX	0.29	LEXP/TEXP	0.13	little influence; strong
CCPLX	0.24			correlation with OXEP/DEEXP
OCAP/DECAP	-0.19	PCON	0.04	low number of different data points
OXEP/DEEXP	-0.36	$\frac{Size_{Reused}}{Size_{Total}}$	-0.10	little influence

Table 3. Selection of Relevant Cost Drivers using Correlation Analysis

drivers in order to get more accurate results. The cost driver **DATA** is strongly correlated with the cost driver **DCPLX**. Most of the surveyed projects took place at one site resulting in limited information about the actual influence of the **SITE** parameter, which was therefore excluded. The cost driver **DOCU** highly correlates with the **OE** cost driver and has only moderate influence on the effort. A similar line of reasoning applies to the cost drivers **LEXP/TEXP** which are highly correlated with **OXEP/DEEXP** while modestly contributing to the prediction variable. The surveyed projects did not experience a permanent personnel turnover, resulting in a very low correlation coefficient for the cost driver **PCON**. Intriguingly, reusing ontologies had only a limited effect on the ontology building effort as indicated by the small negative correlation between $\frac{Size_{Reused}}{Size_{Total}}$ and the effort. Most interviewees reported major difficulties translating and modifying reused ontologies, which obviously offset most of the time savings expected from ontology reuse. The cost driver **TOOL** was not considered in the calibration, because it did not differentiate the projects (i.e. all data points utilized only ontology editors).

The exclusion of the mentioned cost drivers from the current ONTOCOM calibration does not mean, that those cost drivers are not relevant for predicting the ontology building effort. With the currently available data set it is, however, not possible to provide accurate estimates for these cost drivers. The prediction quality for multi-site developments and projects with a high personal turnover might suffer from the exclusion of the corresponding drivers. However, the accuracy of the prediction for the remaining cost drivers increases.

Calibration Results The approximation of the effort multipliers with the linear regression approach implies a reformulation of equation 1. After applying the logarithm function and introducing the parameters β_i as exponents for the cost drivers we ob-

tained the equivalent equation 3.[10] β_i are scaling factors by which the existing effort multipliers should be scaled in order to fit the model. We recall that α is a learning rate factor also used to model economies of scale.

$$\ln(PM_X) = \ln(A) + \alpha * \ln(Size_X) + \sum \beta_i * ln(CD_{Xi}) \tag{3}$$

The linear regression delivers a best fit for the effort multipliers w.r.t. to the surveyed empirical data. However, the relatively small sample size results in a limited accuracy of the estimated effort multipliers. This drawback can be overcome with the help of the a-priori estimations of the parameters, which were defined by human experts. A linear combination of expert estimations and historical data is, however, sub-optimal. The combination should take into account the number of data points used for the linear regression and the variance observed in the expert ratings as well as in the data points. A mulitplier which all experts have given the same rating, while the linear regression results in a high variance should be influenced less by the data than by the experts. Bayesian analysis is a way to achieve the desired outcome [4].

$$\beta^{**} = [\frac{1}{s^2}X'X + H^*]^{-1} \times [\frac{1}{s^2}X'X\beta + H^*b^*] \tag{4}$$

Equation 4 delivers the estimations of the scaling factor β^{**} combining expert knowledge and empirical data in a statistically sound way. s^2 is the variance of the residual data of the sample; X is the matrix of observations; and H^* and b^* is the inverse of the covariance matrix and the mean of the expert estimations, respectively. Figure 2 exemplifies the approach. The lines depict the probability distribution of the productivity range estimations for the expert judgement, the data analysis and the Bayesian combination, respectively. The arrows point to the corresponding means. We note that the experts judgement indicates a productivity range for the cost driver CCPLX of 6.17 with a small variance. Estimating the productivity range based on the data results in a mean of 7.05 with a higher variance, though. The Bayesian analysis induces a shift of the estimation towards the data-driven estimation, but only with a small fraction because its higher data variance.

Table 4 summarizes the results of the Bayesian analysis. In column *Correlation with PM* we list the correlation coefficients for the reduced number of cost drivers with the effort in person months (PM). In the *Significance* column we plot the confidence level for the estimation. Not all effort multipliers could be determined with the same accuracy. A lower confidence level indicates a better estimation. The calibration is very good for, for instance, the exponent α (**SIZE**), but less accurate for the effort multipliers related to **OCAP/DECAP**. The *Productivity range* column lists the relative influence a cost driver has on the final prediction.

[10] This step is only possible if the data is distributed exponentially, thus we have significantly more data points with a low number of entities than with a high number of entities. This holds true for the collected data.

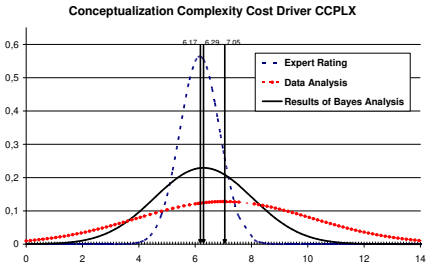

Fig. 2. Productivity Range: Conceptualization Complexity

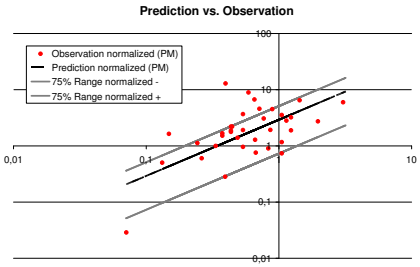

Fig. 3. Comparison of Observed Data with Predictions

Cost Driver	Correlation with PM	Significance	Productivity range
SIZE	0.50	0.001	$\alpha = 0.5$
OE	0.44	0.034	4.0
DCPLX	0.39	0.063	3.2
REUSE	0.38	0.528	5.2
CCPLX	0.24	0.311	6.3
OXEP/DEEXP	-0.36	0.060	1.5
ICPLX	0.29	0.299	0.6
OCAP/DECAP	-0.19	0.925	1.5

Table 4. Statistical Data and Productivity Range of the Effort Multipliers

Based on the results of the calibration Figure 4.2 compares the predictions from the calibrated model with the observations. In order to visualize the results we have normalized the data with the product of the corresponding cost drivers. The gray lines indicate a range around the prediction adding and subtracting 75% of the estimated effort. 75% of the historical data points lie within this range. For the corresponding 30% range the model covers 32% of the real-world data. This indicates a linear behavior of deviation which we consider quite accurate for a very first model. Our goal is that 75% of the data lie in the range of adding and subtracting 20% of the estimated effort.

Discussion of the Calibration Results Although any prediction model provides solely an approximation of the true building efforts, this calibration is already helpful to get an early impression on the expected values. Experiences with cost estimation models in established engineering disciplines suggest that a calibration for a particular company or project team yields more accurate estimations than a general-purpose calibration. Our calibration may therefore predominantly serve as an example for a more context-specific calibration process and may help to identify the resource-intensive activities in a generic ontology engineering process. Moreover, project teams can compare their estimations against a general average value as provided by us. Note also that a calibration uses historical data to estimate future outcomes. Although the average and the variation observed in the historical data may remain constant in future projects, the predicted effort for any specific project may still significantly differ from the actual effort.
Regarding the quality of our model w.r.t. the calibration accuracy it is important to note

that the estimations for the cost drivers **OCAP/DECAP** and **REUSE** have a low significance. For the cost drivers **OCAP/DECAP** this leaves room for improvement, as the data analysis counterintuitively suggests that more capable project teams need longer to develop an ontology. We obtained the same result for the cost driver **OEXP/DEEXP**. The main reason for this artefact may be the fact that ontology engineers from academia were more experienced, implying that they invested more time in developing ontologies than people from industry, whose mode of operation might have been motivated by different factors as in academic projects.
Another interesting finding of the analysis is the relative importance of the cost drivers **Ontology evaluation (OE)**, **Domain complexity (DCPLX)** and **Conceptualization complexity (CCPLX)** in correlation with the observed significance. This indicates that any facilitation in those areas may result in major efficiency gains w.r.t. the overall ontology engineering effort. Moreover, the very high learning rate indicates that the building of very large ontologies is feasible from an economic point of view, although we admit that the number of data points for ontologies larger than 1.000 ontology entities is comparatively low.

5 Related Work

Cost estimation methods have a long-standing tradition in more mature engineering disciplines such as software engineering or industrial production [1, 7, 15]. Although the importance of cost issues is well-acknowledged in the community, as to the best knowledge of the authors, no cost estimation model for ontology engineering has been published so far. Analogue models for the development of knowledge-based systems (e.g., [5]) implicitly assume the availability of the underlying conceptual structures. [9] provides a qualitative analysis of the costs and benefits of ontology usage in application systems, but does not offer any model to estimate the efforts. [3] presents empirical results for quantifying ontology reuse. [8] adjusts the cost drivers defined in a cost estimation model for Web applications w.r.t. the usage of ontologies. The cost drivers, however, are not adapted to the requirements of ontology engineering and no evaluation is provided. We present an evaluated cost estimation model, introducing cost drivers with a proved relevance for ontology engineering, which can be applied in the early stages of an ontology development process.

6 Conclusion

The application of ontologies in commercial applications depends on the availability of appropriate methodologies guiding the ontology development process and on methods for an effective cost management. We propose a parametric cost estimation model for ontologies by identifying relevant cost drivers having a direct impact on the effort invested in ontology building. We evaluate the model a-priori and a-posteriori.

The a-priori evaluation shows the validity of the approach to cost estimation and the meaningful selection of the cost drivers. The a-posteriori evaluation results in high quality estimations for the learning rate α and the cost drivers related to the ontology evaluation and the requirements complexity. These are also among the more relevant cost drivers. Provision of tool support for these two areas of ontology engineering may

thus be particularly effective to facilitate the ontology engineering process. The collection of data will continue towards a more accurate calibration of the model. In particular we intend to approach the suggestions received during the a-priori evaluation w.r.t. a more differentiated size parameter and w.r.t. the support for ontology reuse activities on the basis of a larger number of data points. In the near future we also plan to make the results of our survey public and to provide a service which offers on-the-fly cost estimations for ontology engineering processes based on the available data.

Acknowledgements This work has been partially supported by the European Network of Excellence "KnowledgeWeb-Realizing the Semantic Web" (FP6-507482), as part of the KnowledgeWeb researcher exchange program **T-REX**, and by the European project "Sekt-Semantically-Enabled Knowledge Technologies"(EU IP IST-2003-506826) and "NeOn - Lifecycle Support for Networked Ontologies" (EU IP IST-2005-027595). We thank all interviewees for the valuable input without which this paper could not have been produced.

We encourage the community to participate and contribute experiences from ontology engineering projects at `http://ontocom.ag-nbi.de`.

References

1. B. W. Boehm. *Software Engineering Economics*. Prentice-Hall, 1981.
2. G. Box and G. Tiao. *Bayesian Inference in Statistical Analysis*. Addison Wesley, 1973.
3. P. R. Cohen, V. K. Chaudhri, A. Pease, and R. Schrag. Does Prior Knowledge Facilitate the Development of Knowledge-based Systems? In *AAAI/IAAI*, pages 221–226, 1999.
4. S. Devnani-Chulani. *Bayesian Analysis of the Software Cost and Quality Models*. PhD thesis, Faculty of the Graduate School University of Southern California, 1999.
5. A. Felfernig. Effort Estimation for Knowledge-based Configuration Systems. In *Proc. of the 16th Int. Conf. of Software Engineering and Knowledge Engineering SEKE04*, 2004.
6. A. Gomez-Perez, M. Fernandez-Lopez, and O. Corcho. *Ontological Engineering – with examples form the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Verlag, 2004.
7. C. F. Kemerer. An Empirical Validation of Software Cost Estimation Models. *Communications of the ACM*, 30(5), 1987.
8. M. Korotkiy. On the Effect of Ontologies on Web Application Development Effort. In *Proc. of the Knowledge Engineering and Software Engineering Workshop*, 2005.
9. T. Menzies. Cost benefits of ontologies. *Intelligence*, 10(3):26–32, 1999.
10. National Aeronautics and Space Administration. NASA Cost Estimating Handbook, 2004.
11. E. Paslaru Bontas and M. Mochol. Ontology Engineering Cost Estimation with ONTOCOM. Technical Report TR-B-06-01, Free University of Berlin, January 2006.
12. E. Paslaru Bontas and C. Tempich. How Much Does It Cost? Applying ONTOCOM to DILIGENT. Technical Report TR-B-05-20, Free University of Berlin, October 2005.
13. E. Paslaru Bontas and C. Tempich. Ontology Engineering: A Reality Check. In *5th Int. Conf. on Ontologies, DataBases, and Applications of Semantics (ODBASE2006)*, 2006.
14. G.A.F. Seber. *Linear Regression Analysis*. Wiley, New York, 1977.
15. R. D. Stewart, R. M. Wyskida, and J. D. Johannes. *Cost Estimator's Reference Manual*. Wiley, 1995.
16. Y. Sure, S. Staab, and R. Studer. Methodology for Development and Employment of Ontology based Knowledge Management Applications. *SIGMOD Record*, 31(4), 2002.