Exploring the Economical Aspects of Ontology Engineering

Elena Simperl[1] and Christoph Tempich[2]

[1] STI Innsbruck, University of Innsbruck, Austria, elena.simperl@sti2.at
[2] Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany, tempich@aifb.uni-karlsruhe.de

Summary. A core requirement for the usage of ontologies within enterprizes is the availability of proved and tested techniques which guarantee an efficient engineering of high-quality ontologies, be that by reuse, manual building or automatic knowledge acquisition. Besides feasible technological support this includes in equal measure integrating ontology engineering within the more general framework of enterprize information architectures, and taking into account the economics of ontology engineering projects, in particular issues of cost effectiveness and profitability. This chapter addresses these two aspects. We discuss the role of ontology engineering in the context of enterprize architectures, arguing for the importance of cost-related measures as decision support in planning and controlling. Then we analyze approaches for reliably assessing the costs of building ontologies, and the usage of cost-related information to quantifiably support decisions arising during the life cycle of an ontology and to optimize the operation of associated processes. We account for the similarities and differences between software and ontology engineering in order to establish the appropriateness of applying methods with a long-standing tradition in this adjacent engineering field to ontologies. Building upon the results of this analysis we introduce ONTOCOM as the first cost model for ontologies and discuss different methods to improve its accuracy for a wide range of ontology engineering projects at public and corporate level.

1 Introduction

The dissemination of ontologies and ontology-based applications within enterprizes requires methods and tools which are able to deal with both the technical and the economic challenges of ontology engineering . In order for ontologies to be efficiently built and deployed at large scale one needs mature technologies supporting the entire ontology, as well as proved and tested means to plan and control the overall engineering process as part of more general IT-related processes within an enterprize. A wide range of ontology engineering methodologies have emerged in the Semantic Web community. Apart from minor differences in the level of detail adopted for the description of the process model they define ontology engineering as an iterative process, which shows major similarities to models emerged in the neighbored research field of software engineering. However existing methodologies do not consider the economic factors commonly related to every real-world engineering project, in particular the estimation of development and maintenance costs using predefined cost models, and the impact of such cost information on the operation of an engineering process.

With ONTOCOM we present the first existing approach in this newly emerging field of ontology engineering [17]. Estimating costs for ontology engineering is similar to estimating costs for software (or product) engineering as it requires the consideration of economic aspects for generic products and the processes they result of. Therefore, our approach largely benefits from the experiences made in estimating costs for software engineering. Through expert interviews we identified the most relevant cost drivers for a wide class of ontology engineering projects. In a large user study we acquired relevant data from existing ontology engineering projects and calibrated a parametric cost prediction equation with promising results. Further on, we analyzed the appropriateness of applying alternative approaches such as estimation by analogy and the Delphi method in the same context. Combing the three we were able to identify dimensions for further research and development towards a methodology for the creation of any kind of cost estimation model for ontologies, independently of the ontology life cycle, the cost estimation method, or the organizational setting it might be employed.

Cost estimation methods have a long-standing tradition in more mature engineering disciplines such as software engineering or industrial production [7,9,11,18,20,23]. 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., [8]) implicitly assume the availability of the underlying conceptual structures. Reference [15] 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. Reference [5] presents empirical results for quantifying ontology reuse. Reference [14] 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 early stages of arbitrary ontology development processes and further customized to specific needs of these processes to improve its prediction quality.

The outline of this chapter is as follows. We start by reviewing the economical aspects of ontology engineering in the context of corporate IT, motivating the need for cost estimation models in this field in Sect. 2. Provided reliable means to estimate costs during particular stages of the life cycle of an ontology, Sect. 3 introduces a series of use cases showing how cost information could be utilized to optimize the operation of an ontology-related project. The remaining sections are dedicated to the design and evaluation of a cost model for ontologies. We first analyze potential methods for cost prediction in Sect. 4. Section 5 introduces the ONTOCOM model based on the previously identified most promising methods. Details about its application in concrete ontology engineering projects are provided in Sect. 6. Section 7 summarizes the lessons learned from our research and the planned future work.

2 Economical Aspects of Information Technology

In this section we situate ontology engineering in the IT landscape of an enterprize. We discuss how the development and deployment of ontologies influences the enterprize information architecture and analyze the most important economical aspects related to this setting. As a result we argue for the necessity of reliable instruments for cost prediction for ontology engineering, which is an essential part of the data architecture of an enterprize.

2.1 Enterprize Information Architecture

An enterprize information architecture includes the products, procedures, organizational structures and IT systems of an enterprize. The design of enterprize architectures and their continuously adaptation to new environmental requirements are realized with the help of so-called *architecture development frameworks* such as the Zachmann framework [26] or the TOGAF framework.[1] They provide a comprehensive approach including methodologies, tools, best practices and standardized guidelines to develop a broad range of different IT architectures in an enterprize.

According to the latter an enterprize information architecture is typically modeled iteratively at four levels: Business, Application, Data, and Technology. The design process starts at the business level with the definition of a business strategy, followed by the specification of requirements which lead to an overall architecture vision and to a business, product and process architecture. The following steps are more technically oriented. IT specialists design the application architecture, the actual technology architecture implementing the application, and the *information system/data architecture*, which refers to the data models, or in our case ontologies, used in the application (cf. Fig. 1).

So far the ontology engineering research community has focused mainly on methods and tools for building and managing ontologies without reflecting on the implications arising from enterprize architecture development processes.[2]

[1] `http://www.opengroup.org/togaf/`

[2] An exception from this statement is the IDEF integrated definition method. The IDEF approach defines a function modeling method (IDEF0), information

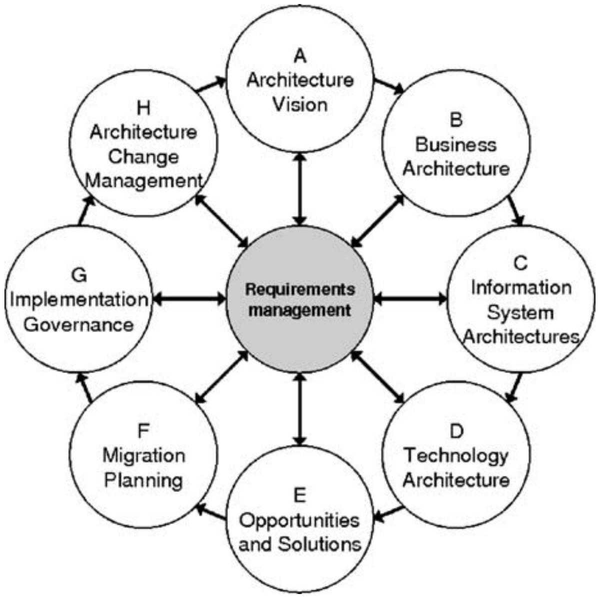
A
Architecture Vision
B
Business Architecture
C
Information System Architectures
D
Technology Architecture
E
Opportunities and Solutions
F
Migration Planning
G
Implementation Governance
H
Architecture Change Management
Requirements management

Fig. 1. Enterprize information architecture according to the TOGAF framework. Ontology engineering is related to (C) Information System Architectures

These implications are twofold: on the one hand the development of an enterprize information architecture results, among other things, in a series of domain and functional requirements regarding the scope and the utilization of the ontology; on the other hand this imposes non-functional boundaries such as the maximal development effort to be invested in the realization of the ontology. Consequently the requirements in the first category need to be matched to the estimated costs related to the development and the maintenance of the ontology.

2.2 Governance

A second essential aspect in the context of an enterprize IT infrastructure is the operation and maintenance of the underlying IT systems, tasks typically accomplished according to a governance framework evaluated and approved by dedicated institutes such as COBIT.[3] Governance frameworks cover the organization, control, steering and diffusion of corporate IT system development:

modeling method (IDEF1), data modeling method (IDEF1X), process description capture method (IDEF3), object oriented design method (IDEF4), and finally the ontology description capture method (IDEF5), which can be mapped to the aforementioned architectural models foreseen by enterprize architecture development frameworks. More information about IDEF is available at `http://www.idef.com/`

[3] `http://www.itgi.org/`

Organization Organizational aspects describe the different roles relevant for the development of an IT system in an organization, their responsibilities and decision procedures.

Steering Steering includes the definition of processes and activities in which the participants act in order to achieve the overall goals. The handling of intellectual property rights is specified as well in this step. This is an important aspect for ontology engineering as a systematic use of ontologies is an important pre-requisite for achieving application interoperability.

Control Control covers the definition of indicators in order to monitor the processes defined in the previous step and be able to detect unintended consequences of a system development or operation. From an ontology engineering perspective this step includes metrics for the characterization of ontologies and the associated development and maintenance processes.

Diffusion Finally, one of the most important aspects in large organizations is the definition of appropriate roll-out mechanisms in order to guarantee that the whole organization is able to follow the proposed processes.

In our work we focus on cost effectiveness, as one of the most important indicators at controlling level in an governance framework. Obviously the costs associated to the development of an IT system, of which ontology engineering is an important part, should be lower than the benefit expected to be obtained through its deployment. The estimation of the efforts related to engineering an ontology is crucial for the planning of data architecture change projects. Taking into account the operation of IT systems, it is worthwhile to pay attention to the reusability of an ontology. A reusable ontology is likely to be more user-friendly, thus reducing the training effort required in the roll-out phase of the associated IT system, while typically involving additional effort in the development. For monitoring purposes it is furthermore interesting to compare the estimated and the actual effort values in order to judge the ability of the organization to develop ontologies.

Provided these various usage scenarios, we argue for the necessity of extending existing IT-specific cost estimation approaches towards ontologies. In the next section we provide a second motivating scenario for this requirement, explaining how knowledge about the efforts implied by the development, maintenance and deployment of an ontology influences its life cycle.

3 Usage of Cost Information During the Life Cycle of an Ontology

The previous section was concerned with the relevance of cost information in the IT organization from a management perspective. This section focuses on the implementation side of ontologies, and thus on the relevance of cost information from a development and maintenance perspective. Figure 2 gives an overview how cost information is typically used within the life cycle of a

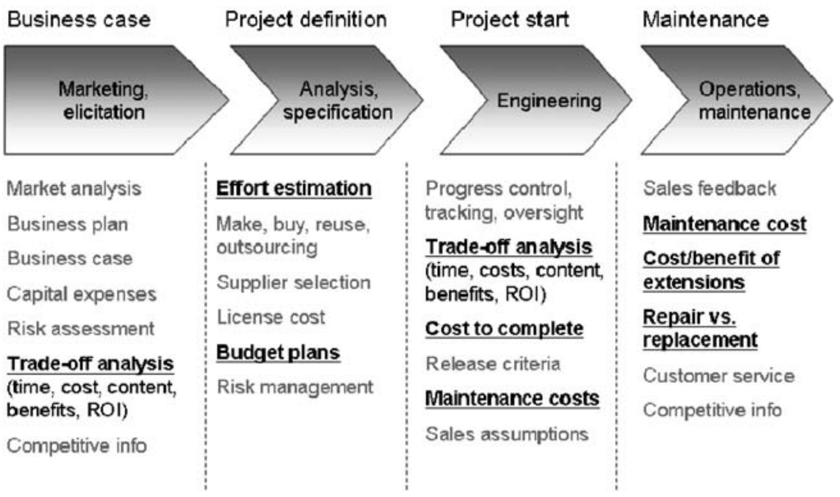

Fig. 2. Cost indicators during the life cycle of a product cf. [6]

product. Most product development projects start with the elaboration of a business case, of which the trade-off analysis is a major part. The trade-off analysis compares the expected benefit with the expected development costs. Turning back to IT systems, in order to make valid decisions based on the trade-off analysis, an accurate estimate of the expected costs and benefits of ontologies within the context of a particular application scenario is thus required in such early phases of a project.

The costs may be estimated with methods proposed in this chapter. The quantification of the benefits are covered by recent proposals to value IT development projects[4] which could be transferred to ontology development.

The effort related to ontology engineering is relevant in the initial project *planning* phase. In this phase the project manager assigns available resources to the planned tasks, taking into account the estimates of the effort associated to them, which are crucial for an on-time delivery of the project outcomes. These estimates are updated during the remaining project phases, as at the beginning of the project information about available skills and other influencing factors might not be available or reliable. Re-estimations during the project allow for adjustments in the project plan and provide a basis for the calculation of the total resources necessary to complete the project. In terms of ontology engineering the availability of cost information helps to make decisions related to the expected quality, size and granularity of the ontology to be developed. This is particularly of interest for ontology engineering methodologies following an iterative or rapid prototyping approach.

Effort estimates can be used in equal measure for *controlling* and *benchmarking* purposes. As such estimates are derived from previous project

[4] `http://www.isaca.org/Content/ContentGroups/Val_IT1/Val_IT.htm`

experiences, the comparison between planned and actual effort values is a benchmark against previous projects, either external or internal. If the variation exceeds certain thresholds the project manager can take early countermeasures or, at least, can thoroughly examine the project and detect the underlying reasons for the variation.

Towards the end of the ontology life cycle effort estimations for ontology *maintenance* can support repair versus replacement decisions. This of course requires knowledge about the total cost of ownership of an ontology and about the cost statements with respect to particular ontology management activities.

To summarize cost estimation models for ontologies are necessary in order to align the discipline of ontology engineering with common IT practice within enterprizes, as well as for shaping the life cycle of ontologies in an economically based fashion. The remaining sections describe how such cost models can be designed, evaluated and used in an organization.

4 Design of an Ontology Cost Estimation Model

In order to design a cost estimation model for ontologies we can resort to established approaches in the field of software measurement, which describe the steps required for this purpose and the way they can be effectively performed within a company (cf. Fig. 3).

Extract In this step the engineering team identifies the cost factors and decides upon the method(s) used to generate the estimates.

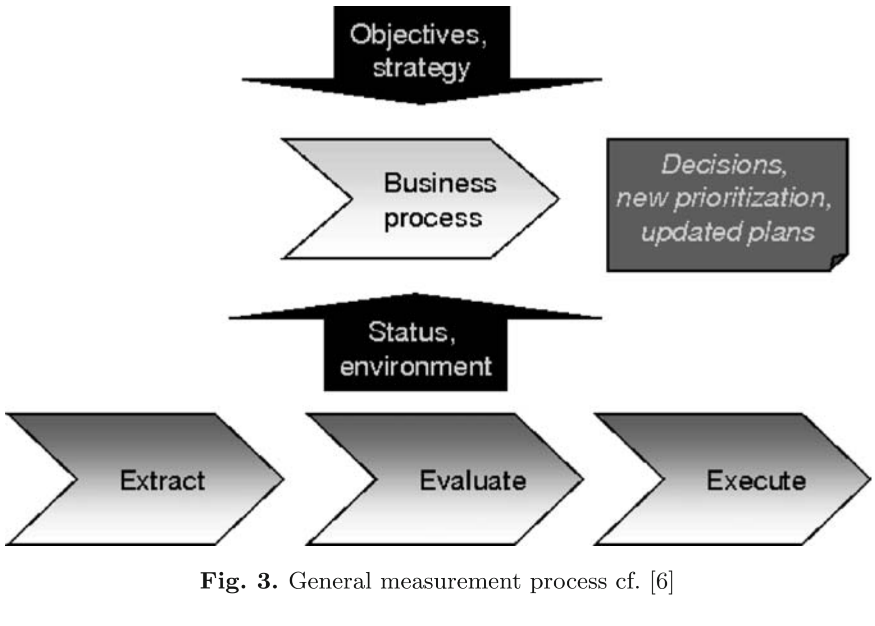

Fig. 3. General measurement process cf. [6]

Evaluate The model previously defined is evaluated and adapted according to a specific procedure (see below). For expert-based methods, the evaluation includes evaluation sessions among the participants. In case of mathematical prediction equations the evaluation uses data collected for this purpose from previous projects. In the latter case it is essential that the evaluation relies on a sufficient amount of historical data *from internal projects* in order to customize the model to the particularities of a given enterprize or department.

Execute Once a feasible quality of the predictions has been achieved the model is used at various stages of the life cycle of a product (in our case an ontology) and in relation to the more general enterprize architecture development process. In this context it is important that the employees understand the necessity of this additional workload and that they are trained to correctly use the model.

We now turn to a description of the first step. Information about the validation of a particular model (in our case based on a parametric approach) are available in [17].

4.1 Generic Methods

Estimating costs for engineering processes can be performed according to several methods, often used in conjunction in order to avoid individual limitations [1, 22].

Expert judgment/Delphi method The Delphi method 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 major advantage of the approach. Its most extensive critique point is related to the difficulties to explicitly state the decision criteria used by the contributing experts and to its inherent dependency of the availability of experts to carry on the process.

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, and depends on the accuracy in establishing real differences between completed and current projects.

Decomposition method This method involves generating a work breakdown structure, i.e., breaking a product into smaller components or a project into activities and tasks in order to produce a lower-level, more detailed description of the product/project at hand, which in turn allows more accurate cost estimates. The total costs are calculated as average values, possibly adjusted on the basis of the complexity of the components/tasks considered. The successful application of the method depends of the availability of necessary information related to the work breakdown structure.

Parametric/algorithmic method This method involves the usage of mathematical equations based on research and historical data from previous projects. The method analyzes main cost drivers of a specific class of projects and their dependencies and uses statistical techniques to refine and customize the corresponding formulas. As in the case of the analogy method the generation of a proved and tested cost model using the parametric method is directly related to the availability of reliable and relevant data to be used in calibrating the initial core model.

Orthogonally to the aforementioned methods we mention two high-level approaches to cost estimation (cf. Table 1).

Bottom-up estimation This approach involves identifying and estimating costs of individual project components separately and subsequently summing up the outcomes to produce an estimation for the overall project. As such the bottom-up approach is at the core of the decomposition method introduced above.

Top-down estimation By contrast the top-down method relies on overall project parameters. For this purpose, the project is partitioned into lower-level components and life cycle phases beginning at the highest level. The approach produces are total project estimates, in which individual process tasks or product components are responsible for a proportion of the total costs.

The decomposition method is based on a bottom-up approach. Estimation by expert judgment, analogy or parametric equations can be carried out in a top-down or a bottom-up fashion, also depending of the stage of the project in

Table 1. Methods and approaches to cost estimation

	Bottom-up estimation	Top-down estimation
Expert judgement method	Experts estimate the costs of low-level components or activities	Experts estimate the total costs of a product or a project
Analogy method	Costs are calculated using analogies between low-level components or activities	Costs are estimated using a global similarity function for products or projects
Decomposition method	Costs are calculated as an average sum of the costs of lower-level units, whose development effort are known in advance	Not applicable
Parametric method	Costs are calculated using a statistic model which predicts the costs of lower-level units on the basis of historical data about the costs of developing such units	Costs are calculated using a statistic model which is calibrated using historical data about, and predicts the current value of the total development costs

which the estimates need to calculated. Top-down estimation is more applicable to early cost estimates when only global properties are known, but it can be less accurate due to the less focus on lower-level parameters and technical challenges usually predictable later in the process life cycle, at most. The bottom-up approach produces results of higher-quality, provided a realistic work breakdown structure and means to estimate the costs of the lower-level units the product/project has been decomposed into.

4.2 Methods Feasible for Ontology Engineering

In the following we examine the advantages and disadvantages of each of the aforementioned approaches given the product- and process-related characteristics of ontology engineering and the current state of the art in the field:

Expert judgment/Delphi method The expert judgement method seems to be appropriate for our goals since large amount of expert knowledge with respect to 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.

Analogy method The analogy method requires knowledge about the features of an ontology, or of an ontology development process, which are relevant for cost estimation purposes. Further on it assumes that an accurate comparison function for ontologies is defined, and that we are aware of cost information from previous projects. While several similarity measures for ontologies have already been proposed in the Semantic Web community, no case studies on ontology costs are currently available. There is a need to perform an in-depth analysis of the cost factors relevant for ontology engineering projects, as a basis for the definition of such an analogy function and its customization in accordance to previous experiences.

Decomposition method This method implies the availability of cost information with respect to single low-level engineering tasks, such as costs involved in the conceptualization of single concepts or in the instantiation of the ontology. Due to the lack of available information the decomposition method can not be applied yet to ontology engineering.

Parametric/algorithmic method Apart from the lack of costs-related information which should be used to calibrate cost estimation formula for ontologies, the analysis of the main cost drivers affecting the ontology engineering process can be performed on the basis of existing case studies on ontology building, representing an important step toward the elaboration of a predictable cost estimation strategy for ontology engineering processes. The resulting parametric cost model has to be constantly refined and customized when cost information becomes available. Nevertheless the definition of a fixed spectrum of cost factors is important for a controlled collection of existing real-world project data, a task which is

Table 2. Cost estimation methods and approaches potentially applicable to ontology engineering

	Bottom-up estimation	Top-down estimation
Expert judgment method	Currently not feasible	Feasible
Analogy method	Currently not feasible	Feasible
Decomposition method	Currently not feasible	Not applicable
Parametric method	Currently not feasible	Feasible

fundamental for the subsequent model calibration. This would also be useful for the design and customization of alternative prediction strategies, such as the aforementioned analogy approach.

Given the fact that cost estimation has been marginally explored in the Semantic Web community so far, and that little is known about the underlying cost factors , a bottom-up approach to the previously introduced methods is currently not practicable, though it would produce more accurate results. In turn, expert judgment, analogy and parametric cost estimates could be obtained in a top-down fashion, if the corresponding methods are clearly defined and customized in the context of ontology engineering. A summary of the results of this feasibility analysis is depicted in Table 2. Due to the incompleteness of the information related to cost issues, a combination of the three methods is likely to overcome certain limitations of single ones.

Section 5 introduces the ontology cost model ONTOCOM and discuss ways to improve its prediction quality. ONTOCOM follows a top-down approach to cost estimation, by identifying the cost drivers associated to the most important phases of the ontology life cycle and calculating a global effort estimate on the basis of different prediction methods. The current version of ONTOCOM investigates the usage of the parametric, the analogy and the Delphi methods to ontology engineering.

5 ONTOCOM: A Cost Model for Ontology Engineering

In this section we introduce the generic ONTOCOM cost estimation model. The model is generic in that it assumes a sequential ontology life cycle, according to which an ontology is conceptualized, implemented and evaluated, after an initial analysis of the requirements it should fulfill (see below). By contrast ONTOCOM does not consider alternative engineering strategies such as rapid prototyping or agile methods, which are based on different life cycles.[5] This limitation has been issued in previous work of ours, which describes how the generic model should be customized to suit such scenarios [16, 21].

[5] Reference [10] for a discussion on the relation between this process model and the IEEE standards [12].

Table 3. Design of the ONTOCOM cost model (parametric, analogy and Delphi methods)

	Parametric method	Analogy method	Delphi method
Extract	Define work breakdown structure		
	Define cost drivers and ratings		Provide individual estimations
	–	Define similarities	
Evaluate	Collect data		Agree on estimates
	Perform statistical calibration	Calibrate weights	
Execute	Specify ratings of the cost drivers which correspond to the application at hand		Calculate final result
	Insert values to the prediction equation		
	Calculate estimate using equation		

The cost estimation model is realized as follows. 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]. Then we can derive the global costs using various methods applicable for this top-down approach (cf. Table 3). Currently we are looking into three methods: the *parametric*, the *analogy* and the *Delphi* method, respectively.

For the parametric method the result of these steps is a statistical prediction model (i.e., a parameterized mathematical formula). Its parameters are given start values in pre-defined intervals, and are subsequently calibrated on the basis of previous project data. This empirical information complemented by expert opinions is used to evaluate and revise the predictions of the initial a priori *model*, thus creating a validated a posteriori *model*. The analogy method works similarly. It is based on a similarity equation, which is a mathematical formula aggregating similarity functions on the basic cost dimensions in a weighed fashion. The weights need to be specified according to empirical calibration and/or expert judgement, just as in the case of the parametric method. The Delphi method can be applied independently of any prediction formula or analogy function (see below).

The parametric equation has been carefully evaluated using statistical calibration and Bayes analysis as described in [17], whilst the analogy one is in the process of being customized. The Delphi method has been applied several times to derive specific initial inputs for the previous two methods.

5.1 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 [10,25] According to them ontology building consists of the following core steps (cf. Fig. 4):

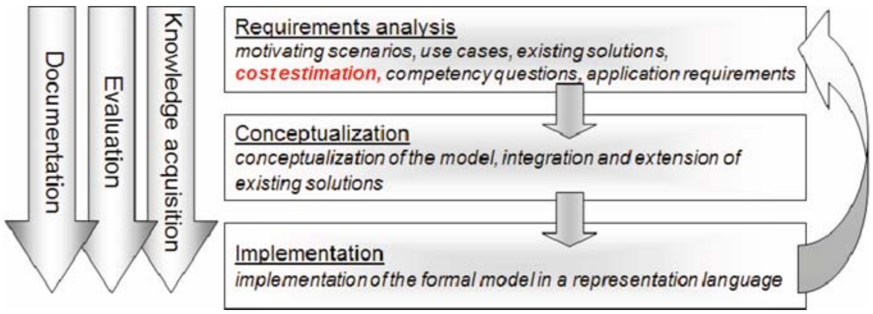
Documentation
Evaluation
Knowledge acquisition
Requirements analysis
motivating scenarios, use cases, existing solutions, cost estimation, competency questions, application requirements
Conceptualization
conceptualization of the model, integration and extension of existing solutions
Implementation
implementation of the formal model in a representation language

Fig. 4. Typical ontology engineering process

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 pre-defined 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 [24]. In particular this contains a set of competency questions describing the domain to be modeled 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.

Conceptualization The application domain is modeled in terms of ontological primitives, e.g. concepts, relations, axioms.

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.

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.

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 [10], 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 [24] 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 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 modeled 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.

We now introduce the cost drivers associated to this work breakdown structure.

5.2 The ONTOCOM Cost Drivers

The ONTOCOM cost drivers, which are proved to have a direct impact on the total development efforts, can be roughly divided into three categories [16,17]:

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 item 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

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

Table 4. The Conceptualization complexity cost driver *CCPLX*

Rating Level	*Description*
Very Low	Concept list
Low	Taxonomy, high nr. of patterns, no constraints
Nominal	Properties, general patterns available, some constraints
High	Axioms, few modeling patterns, considerable nr. of constraints
Very High	Instances, no patterns, considerable nr. of constraints

- 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

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 have been defined after extensively surveying recent ontology engineering literature and conducting expert interviews, and from empirical findings of numerous case studies in the field [16]. 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 4. 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 a final rating/value of the corresponding cost driver by means of normalized weights [16].

When using the model the project manager needs to specifies the current rating level for each cost driver according to the setting to which the estimation applies.

5.3 The Parametric Method

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

$$PM = A * Size^{\alpha} * \prod CD_i \qquad (1)$$

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, 13, 18] 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 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.[6] Third we combined the expert estimations and the results of the linear regression in a statistically sound way using Bayes analysis [2]. More details on the calibration results are available in [17].

5.4 The Analogy Method

The analogy method has several advantages when compared to the parametric one, the most important being probably that its usage in a new measurement environment does not require additional calibration efforts, which potentially lead to varying accuracy levels for particular cost drivers. These advantages come, however, at the cost of significant computational power required to calculate similarities, therefore both methods can be seen as candidate techniques to be applied in conjunction [3].

The analogy method defines similarities for each of the cost drivers associated to the work breakdown structure and cumulates the results linearly in a weighed equation:

$$SIM = min \sum_{i,j=1,n} w_i * sim_i(CD_{i,current}, CD_{i,j}) \qquad (2)$$

[6] 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 [19].

In Equation 2 CD_i are ratings of the cost drivers elaborated above, including the size of the ontology to be built. sim_i is the similarity defined for ratings of the cost driver CD_i. The parameter w_i is the weight for this cost driver, all weights summing up to 1. Typically one uses the Euclidian distance as similarity function. The equation identifies the previous project with the closest values of the cost drivers as compared to the current project, and uses this overall similarity value to compute the estimate. j is an index of the size n of the project data set used for the comparisons.

5.5 The Delphi Method

The Delphi or expert judgement method for cost estimation [1] is suitable for ontology engineering projects in its generic form. Every Delphi process involves a moderator and a decision team of three to seven members, which meet two times in order to provide a consensual solution to a particular problem statement. In our context the experts are provided information about the current ontology engineering project and are asked to deliver an estimate of the development efforts according to their experience.

During the first brainstorming meeting the estimation team agrees upon the work breakdown structure, then the individual members provide estimates for the activities covered by this decomposition. In the second meeting the team aims at achieving a consensus on the final estimation by reviewing and revising the inputs of the members. This is an iterative process led by the moderator according to pre-defined rules. Once an agreement on the activity-based estimates has been achieved, the results are collected and compiled into a global figure, which can be used in the project.

As aforementioned such consensus-driven estimations can be used in combination with other methods and for particular cost drivers or activities in order to adjust the effects of data entries which might be unavailable, unreliable or skewed. For example, we used expert estimations of the productivity range of the ONTOCOM cost drivers for the calibration of the parametric equation [17]. A second important use case for such procedures is the estimation of the size of the prospected ontology, which is a core parameter of statistical methods. In terms of the analogy method, expert opinion is crucial for defining the similarity functions for each cost driver, for assigning a priori value to the weights and for evaluating the overall similarity equation.

6 Using ONTOCOM

Starting from a typical ontology building scenario, in which a domain ontology is created from scratch by the engineering team, we simulate the cost estimation process according to the parametric method underlying ONTOCOM. Given the top-down nature of our approach this estimation can be realized in the early phases of a project. 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.

The first step of the cost estimation is the specification of the size of the ontology to be built, expressed in thousands of ontological primitives (concepts, relations, axioms and instances): if we consider an ontology with 1,000 concepts, 200 relations (including is-a) and 100 axioms, the size parameter of the estimation formula will be calculated as follows:

$$Size = \frac{1{,}000 + 200 + 100}{1{,}000} = 1.3 \tag{3}$$

The next step is the specification of the cost driver ratings corresponding to the information available at this point (i.e., without reuse and maintenance factors, since the ontology is built manually from scratch). Depending on their impact on the overall development effort, if a particular activity increases the nominal efforts, then it should 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. Cost drivers which are not relevant for a particular scenario, or are perceived to have a nominal impact on the overall estimate, should be rated with the nominal value 1, which does not influence the result of the prediction equation.

Assuming that the ratings of the cost drivers are those depicted in Table 5 these ratings are replaced by numerical values. The value of the DCPLX cost driver was computed as an equally weighted, averaged sum of a high-valued rating for the domain complexity, a nominal rating for the requirements complexity and a high effort multiplier for the information sources complexity (for details of other rating values see [17]). According to the formula 1 ($\alpha = 1$) the development effort of 11.44 PM would be calculated as follows:

$$PM = 2.92 * 1.3^1 * (1.26 * 1^{10} * 1.15 * 1.11 * 0.93 * 1.11 * 0.89 * 1.2 * 1.7) \tag{4}$$

The constant A has been set to 2.92 after the calibration of the model, while the economies of scale are so far not taken into consideration.

In order to use ONTOCOM in a particular setting (enterprize, business domain, types of ontologies, to name only a few criteria) the generic model should be customized according to the following steps:

- Refine and adapt the work breakdown structure in the light of the applied life cycle and process model followed when engineering the ontology
- Define the statistical prediction model (i.e., a parameterized mathematical formula)
- Calibrate the a priori model based on previous project data to create a valid (more accurate) a posteriori model
- Use the calibrated model to predict development costs

Table 5. Values of the cost drivers

Cost driver	*Effort*	*Value*	*Cost driver*	*Effort*	*Value*
Product factors			*Personnel factors*		
DCPLX	High	1.26	OCAP	High	1.11
CCPLX	Nominal	1	DCAP	Low	0.93
ICPLX	Low	1.15	OEXP	High	1.11
DATA	High	1	DEEXP	Very Low	0.89
REUSE	Nominal	1	LEXP	Nominal	1
DOCU	Low	1	TEXP	Nominal	1
OE	Nominal	1	PCON	Very High	1
Project factors					
TOOL	Very Low	1	SITE	Nominal	1

An example how the generic ONTOCOM model can be applied to a different ontology engineering methodology is described in [21]. Details about a similar enterprize based however on a particular type of ontologies can be found in [4].

7 Conclusions

Technologies related to the development, deployment and maintenance of ontologies have reached a maturity level that they become relevant for businesses. At this stage ontology engineering can no longer be accounted for in a stand alone manner, but should be integrated into the overall architecture and organization of an enterprize. We have shown how ontology engineering fits into existing architecture development frameworks: ontology engineering is an integral part of the information system architecture and influences the technology architecture of an enterprize. Companies complement their overall architecture with a governance framework setting the rules to organize, steer, control and diffuse its deployment. A major concern of IT governance is to timely identify changes in the architecture which are potentially of benefit and to control the realization of the expected benefits. In this context the availability of cost information related to the engineering of ontologies becomes important both at the beginning of an ontology engineering process and during its operation.

In this chapter we have focused on the estimation of costs related to ontology engineering for planning purposes. We have discussed different methods to derive cost information from the environmental setting an existing knowledge and selected three for a more detailed presentation. Following a top-down approach all methods start with a definition of the work breakdown structure. The Delphi method is based on consensual expert estimates aligned to this work breakdown structure, which are aggregated by the project manager towards a final effort prediction. The parametric and analogy method define cost drivers and rating levels as a basis for the mathematical equations customized according to historical project data.

The results from our case studies point in several directions. On the one hand incorporating cost-related aspects into ontology engineering practice is likely to facilitate the interaction between the ontology engineering community and business people. Cost information allow non-engineers to integrate ontology engineering into their management frameworks and makes the creation of ontologies more transparent from a business perspective. On the other hand the estimations are far from being precise yet. First results imply that the creation of glossary-like structures is well understood and the related effort predictable. By contrast the effort related to the creation of ontologies with a high axiomatization is hardly predictable and the exact correlations remain an open issue for future research.

Hence, we see in number of new research directions for the economics of ontology engineering. From a management perspective open issues remain in the areas of controlling and the applicability of the cost models for non-experts. Tool support and additional training materials are needed to ease non-experts the interaction with these models and to guarantee their correct usage. From a technical perspective, in the near future we intend to continue the data collection procedure in order to improve the quality of the generic model and its customizations. Much work needs to be done by many people, thus we see ONTOCOM as a seed for an urgently needed field of research, the cost estimation for ontologies. Any significant improvement in this field will substantially facilitate the uptake of semantic technologies for industrial projects. A second direction of research is related to the refinement of alternative methods for the estimation of critical input parameters such as the size of the prospected ontology. The analogy method seem to be a promising approach for this purpose.

Acknowledgments

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). Further information about ONTOCOM can be found under: `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. L. C. Briand, K. El Emam, D. Surmann, I. Wieczorek, and K. D. Maxwell. An assessment and comparison of common software cost estimation modeling techniques. In *ICSE '99: Proceedings of the 21st International Conference on Software Engineering*, pages 313–322, Los Alamitos, CA, USA, 1999. IEEE Computer Society.
4. T. Buerger, C. Ammendola, and E. Simperl. Evaluation of the economics of multimedia ontologies (salero deliverable d3.1.4). Technical report, STI Innsbruck, 2008.
5. 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.
6. C. Ebert, R. Dumke, M. Bundschuh, and A. Schmietendorf. *Best Practices in Software Measurement.* Springer, 2005.
7. B. W. Boehm et al. *Software Cost Estimation with COCOMO II (with CD-ROM).* Prentice-Hall, 2000.
8. A. Felfernig. Effort estimation for knowledge-based configuration systems. In *Proc. of the 16th Int. Conf. of Software Engineering and Knowledge Engineering SEKE04*, 2004.
9. L. Fischman, K. McRitchie, and D. D. Galorath. Inside SEER-SEM. *The Journal of Defense Software Engineering*, 2005.
10. 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, 2004.
11. W. S. Humphrey. Using a defined and measured personal software process. *IEEE Software*, 13(3):77–88, 1996.
12. IEEE Computer Society. IEEE Standard for Developing Software Life Cycle Processes. IEEE Std 1074-1995, 1996.
13. C. F. Kemerer. An Empirical Validation of Software Cost Estimation Models. *Communications of the ACM*, 30(5), 1987.
14. M. Korotkiy. On the effect of ontologies on web application development effort. In *Proc. of the Knowledge Engineering and Software Engineering Workshop*, 2005.
15. T. Menzies. Cost benefits of ontologies. *Intelligence*, 10(3):26–32, 1999.
16. E. P. 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.
17. E. Simperl, C. Tempich, and Y. Sure. Ontocom: A cost estimation model for ontology engineering. In *Proceedings of the 5th International Semantic Web ISWC2006*, pages 625–639, Springer, 2006.
18. L. H. Putnam and W. Myers. *Measures for Excellence: Reliable Software on Time, Within Budget.* Yourdon, 1991.
19. G.A.F. Seber. *Linear Regression Analysis.* Wiley, 1977.
20. M. Shepperd, C. Schofield, and B. Kitchenham. Effort estimation using analogy. In *Proceedings of the 18th International Conference on Software Engineering ICSE1996*, pages 170–178, 1996.
21. E. Simperl, C. Tempich, and M. Mochol. Cost Estimation for Ontology Development: Applying the ONTOCOM Model. In *Technologies for Business Information Systems*, pages 327–340. Springer, 2007.
22. A. Stellman and J. Green. *Applied Software Project Management.* O'Reilly Media, 2005.

23. R. D. Stewart, R. M. Wyskida, and J. D. Johannes. *Cost Estimator's Reference Manual*. Wiley, 1995.
24. Y. Sure, S. Staab, and R. Studer. Methodology for development and employment of ontology based knowledge management applications. *SIGMOD Record*, 31(4), 2002.
25. Y. Sure, C. Tempich, and D. Vrandecic. Ontology engineering methodologies. In *Semantic Web Technologies: Trends and Research in Ontology-Based Systems*. Wiley, 2006.
26. J. A. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 26(3), 1987.