Abstract

We present ONTOCOM, a method to estimate the costs of ontology engineering, as well as project management tools that support the application of the method. ONTOCOM is part of a broader framework we have developed over the five years, whose aim is to assess the business value of semantic technologies through a suite of methods, estimation models and project management tools, by which the costs and benefits of the corresponding projects are defined, measured and analyzed. The framework supports the engineering of different types of knowledge structures, including ontologies, taxonomies and folksonomies, and of information management systems leveraging such knowledge structures. It also includes benefit analysis models whose results can be used in conjunction with cost-related information in order to identify potential cost savings and to assess the feasibility of specific engineering strategies, in particular ontology reuse. The application of the methods proposed in the framework is supported by project management tools which can be used to customize these methods to a given project environment, to evaluate and validate the underlying estimations using empirical data, and to take into account their results for planning and controlling purposes.

Keywords: ontology engineering, ontology economics, cost estimation, planning and controlling
1. Introduction

In the last years we have seen a continuous uptake of semantic technologies - most recently on the open Web, driven by the Linked Data movement, but in equal measure also in enterprise environments. The key distinct feature of semantic computing compared to other information management technologies is their use of Web standards - languages for knowledge representation, as well as protocols for exposing, accessing and exchanging this knowledge - to structure and formalize information in a way that enables computers to 'understand' complex concepts and situations in a similar way humans do. Ontologies, defined as reusable models capturing the knowledge in a given domain, are one of the core building blocks of the semantic-technologies stack; in combination with components for semantic data management, reasoning, search, as well as annotation of digital artifacts, they can facilitate the development of sophisticated and economically feasible solutions to many prevailing problems in today’s information management.

Probably one of the best showcases for the added value of ontologies is GoodRelations, an ontology about business data and eCommerce transactions. With GoodRelations companies can augment the description of their offerings with structured, machine-processable information, and such enhanced descriptions are leveraged by search engines and other systems to improve the quality of their information management algorithms - matchmaking, filtering, ranking or recommendation - and potentially increase the visibility of the company in response to customer needs. In areas such as, for instance, life sciences, and media and publishing the prospects for using ontologies and other complementary semantic technologies look equally promising, both in terms of the types of functionality, applications and services these technologies enable, and the economical feasibility of their deployment and maintenance.

The research presented in this article substantiates this very last aspect. We introduce ONTOCOM, a method to estimate the costs of ontology engineering projects. ONTOCOM is part of a broader framework we have developed, whose aim is to assess

\[\text{http://purl.org/goodrelations/}\]
the business value of semantic technologies through a suite of methods, cost/benefit estimation models and project management tools, by which the costs and benefits of the corresponding projects are defined, measured and analyzed. The framework supports the engineering of different types of knowledge structures; not just ontologies in the classical sense, but also lightweight structures such as taxonomies and folksonomies, as well as information management systems leveraging them. It also includes benefit analysis models whose results can be used in conjunction with cost-related information in order to identify potential cost savings and to assess the feasibility of specific engineering strategies, in particular ontology reuse. The application of the methods proposed in the framework is supported by project management tools which can be used to customize these methods to a given project environment, to evaluate and validate the underlying estimation models using empirical data, and to take into account their results for planning and controlling purposes. Besides project managers, our work targets CTOs and CIOs of companies interested in an investment in semantic technologies. It offers a systematic and proven-and-tested means to argue in favor of the economical feasibility of these technologies, beyond the evidence available through a number of experience reports on cost savings published during the last years, which may not generalize.

In this article we focus on the most important component of our framework, the ONTOCOM method, and on the project management tools that support its application in ontology engineering projects. We first discuss how cost/benefit information can shape and optimize the operation of ontology engineering projects (Section 2). Then we introduce the actual cost estimation method in terms of the overall approach to cost estimation it follows, the features of ontology engineering projects that are expected to have an impact on the total development costs, and the underlying cost estimation model (Section 3). In Section 4 we present the methodology we used to calibrate the model in two iterations based on 36 and 148 data points, respectively, and analyze the evaluation results. In Section 5 we briefly explain how ONTOCOM can be used in a set of representative ontology engineering scenarios, and present the project management tools that have been implemented for this purpose. The paper is concluded with a wrap-up of the related work (Section 6) and a discussion of future directions of research and
development in the field of ontology economics (Section 7).

2. Ontology Economics as Part of the Ontology Life Cycle

Ontology engineering is defined as ‘the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies’ [14]. The life cycle of an ontology is composed of different activities that can be roughly classified into initialization (such as business case development, feasibility studies, and overall project definition), development, and maintenance activities - just as in other design and engineering disciplines. Furthermore, the life cycle includes a series of management activities such as controlling, planning, and quality assurance, which are orthogonal to the actual production process. Over the past decades the ontology engineering literature has investigated each of these activities and classes of activities in great detail, and has provided extended assistance in form of methodologies, guidelines, best practices, techniques and software environments to support the operation of ontology engineering projects.

The way an ontology engineering project is planned and executed, including the activities discussed above, but also other factors related to the capabilities and expertise of the team and the overall setting which frames the actual project, crucially determines the costs and the quality of the resulting ontology. In an enterprise environment the development of any artifact, and the process and procedures that the engineering team follows to do so, are tightly coupled with pre-defined quantitative (such as low costs of development, maintenance and deployment, and market share increases) and qualitative goals (such as high user satisfaction). Cost/benefit analysis is one of the instruments used in decision-making to identify the business potential of new products and technologies, and to prevent recurring issues related to unrealistic or uncontrolled planning.

Our work is particularly concerned with the use of cost/benefit information along the life cycle of ontologies in order to improve the overall outcomes of ontology engineering projects by offering quantitative means to estimate, supervise and plan the resources required to successfully complete such projects. In the initialization phase
of a project, which includes business case and strategy development, the projections delivered by methods such as ONTOCOM may support trade-off analysis according to (development) time, costs, content, benefits, and return of investment (ROI). In this case, cost/benefit information will most likely be used by senior management planning a new business or adopting a new technology. Such information may yield insights that are useful to decide whether to introduce an ontology-based application at all, and to appraise the ROI of the development of a particular product feature leveraging semantic technologies. In the project and concept definition phase information about the costs and benefits of ontology-based technologies is typically included in effort calculations as part of budget plans, supporting buy vs reuse decisions. During development and market entry managers are interested in estimating the costs to completion of a certain product or project, in controlling the actual and planned costs, and in taking corrective measures in the development process where necessary. Finally, in the operation and maintenance phase methods to study and measure costs and benefits may prove useful to decide for repair against replacement, and for the budgetary planning and control of the overall maintenance. To effectively support management decisions it is necessary to align ontology engineering with common IT practices within enterprises, and to revise ontology engineering methodologies so that the actual development of an ontology can be influenced by economical rationales in addition to the content and technical-related matters that are taken into account by the state-of-the-art in the field.

The following sections introduce ONTOCOM and related extensions that can be used to derive quantitative and qualitative statements with respect to the costs and benefits of ontology-based applications. An account is given on how the method can be applied in a series of project scenarios and how it can be aligned to process-oriented methodologies typically followed in these project. A more detailed description of these scenarios is available in [42]. There we also explain how ONTOCOM results can be used within state-of-the-art ontology project planning, scheduling, and development tools as part of the NeOn Toolkit.²

²http://www.neon-toolkit.org/
3. The ONTOCOM Method

In this section we introduce the core of the ONTOCOM cost estimation method. The method is based on a simplified, sequential version of the ontology life cycle, according to which an ontology is conceptualized, implemented and evaluated after an initial analysis of the requirements it should fulfill.

The cost estimation method is devised 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, 25]. Then a cost model is developed and calibrated following a 'parametric' approach [1, 25]. The parametric approach identifies and analyzes so-called cost drivers, which correspond to characteristics of the class of engineering projects to which the method will be applied, and are presumed to have an impact - positive or negative - on the total costs. Each cost driver is associated to a parameter in the actual cost estimation model (see Section 3.3). The values of the parameters are initialized by ontology engineering experts within pre-defined intervals, and are refined based on statistical techniques on real-world data. Finally, the calibrated model is evaluated with respect to the accuracy of its estimates on empirical historical data - obviously a new data set independent of the one used for the calibration - and in ongoing projects as part of planning and controlling activities. The approach followed by ONTOCOM is not specific to ontology engineering, or in fact to any other related disciplines, most notably software engineering; nevertheless software engineering, and in particular the work around methods such as COCOMO [1, 10] offer interesting insights about the strengths and weaknesses of the parametric approach in particular with respect to its applicability to different development life cycles shared by both ontology and software engineering - such as waterfall, spiral and agile development - and the usage of specific statistical techniques [10].

The history of ontology engineering methodologies has been constantly attesting the commonalities of the two disciplines, and ontology engineering methodologies have learned from their software counterparts in terms of scientific methodology, specific methods and techniques, tool support, and quality of process-oriented descriptions and guidelines. Research on ontology engineering methodologies such as Methontol-
parametric cost estimation, performed a survey and interviewed ontology-engineering experts in order to come up with a preliminary list of cost driver, and took experiences and results from software engineering into account when applying specific techniques to calibrate the parameters based on expert-defined and empirical data.

Other top-down cost estimation methods, most notably the analogy method [1], could be equally employed to come up with an ontology-engineering-specific cost model. The analogy method works with a similarity equation that aggregates in a weighed fashion similarity measures defined on cost drivers. The weights need to be specified via empirical calibration and/or input from experts, just as in the case of the parametric approach. The cost dimensions are subject to a similar analysis as we conducted for ONTOCOM the shared aim of the two methods being the identification of those features of ontology-engineering projects that are presumed to influence the total development costs. A more detailed analysis of the applicability of different generic cost estimation approaches to the area of ontology engineering is available in [38].

In the following we describe the three steps we followed to devise the ONTOCOM method and their results.

3.1. The Work Breakdown Structure

The top-level organization of a generic ontology engineering process can be determined through an analysis of the available process-driven methodologies in the field [8, 14, 32]. In accordance to these methodologies ontology engineering can be divided into the following core steps:

Requirements Analysis The engineering team consisting of domain experts and ontology engineers undertake an analysis of the general project setting in order to identify and prioritize requirements that the ontology to be developed needs to
meet. This step typically includes knowledge acquisition activities by which existing information sources are evaluated with respect to their relevance for the project at hand. These knowledge resources may vary in terms of their structure (and ‘structuredness’), level of formality and topics they cover. The techniques applied to transform, customize and integrate them into a final ontology (in the conceptualization and implementation steps, see below) are equally diverse. The result is an ontology requirements specification document [40], which 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 participants and the engineering methodology. As we will see later on, the availability of a requirements specification is essential for any subsequent cost estimation exercise, as such a specification yields details of the values to be assigned to key parameters of the cost estimation method.

**Conceptualization** The application domain is modeled in terms of ontological primitives such as classes, attributes, relationships, and instances. This step may also include re-engineering activities by which the conceptual models of external information sources are extracted and aligned with those parts of the ontology that is developed from scratch.

**Implementation** The conceptual model is implemented in a knowledge representation language, whose expressivity is appropriate for the richness of the conceptualization and the technology which will utilize the ontology for various information management tasks. If required reused ontologies and those generated from other information sources are translated into the target representation language, merged, and integrated.

**Evaluation** The ontology is evaluated against the set of competency questions contained in the requirements specification. The evaluation may be performed automatically, if the competency questions are represented formally, or semi-automatically using specific heuristics and human judgement. The result of the evaluation is
reflected in a set of modifications and refinements at the requirements, conceptualization and implementation levels.

**Documentation** The ontology, as well as the full footprint of the engineering activities performed is documented to facilitate subsequent reuse and maintenance.

![Ontology Engineering Process Diagram]

Figure 1: Ontology Engineering Process

Depending on the ontology life cycle underlying the process-driven methodology, the aforementioned steps are ordered as a sequential workflow or parallel activities. Methontology [14], which applies prototypical engineering principles, considers *knowledge acquisition, evaluation and documentation* as being complementary support activities performed in parallel to the main development process that essentially consists of requirements gathering, conceptualization and implementation (see Figure 1). Other methodologies, usually following a classical waterfall model, consider these support activities as part of a sequential engineering process, in a similar manner as the one we have just briefly discussed. The OTK-Methodology [40] introduces an initial *feasibility study* in order to assess the risks associated with an ontology engineering enterprise. Other optional steps are *ontology population/instantiation* and *ontology evolution/maintenance*. The former deals with the alignment of application data to the implemented ontology, for instance by extracting information from textual documents and labeling it with ontological concepts. The latter relates to modifications of the ontology performed according to new application and usage requirements, updates of the reused sources, and changes in the modeled domain. Just as in any other engineering discipline, reusing existing sources - in particular ontologies and similar forms
of knowledge organization such as thesauri, taxonomies and classifications - is a central topic of any process-driven ontology engineering methodology. In the simplified process model according to which we defined the work breakdown structure underlying ONTOCOM, reuse is understood as complementary to manual development (see Figure 2). For our cost estimation method we assume that relevant ontologies and ontology-like structures are available to the engineering team during the requirements analysis phase. Subsequently, reuse consists of a series of ontology assessment and customization activities [29, 33]. Ontology engineers and domain experts get familiar with the corresponding ontologies, and judge their relevance for the application scenario at hand. Customization covers translation, segmentation, as well as merging and integration of the reuse candidates into the final ontology.

![Figure 2: Ontology Reuse as Part of Ontology Engineering](image)

We now introduce the cost drivers associated to the work breakdown structure discussed in this section.
### Table 1: The Conceptualization Complexity Cost Driver CCPLX

<table>
<thead>
<tr>
<th>Rating Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>concept list</td>
</tr>
<tr>
<td>Low</td>
<td>taxonomy, high number of patterns, no constraints</td>
</tr>
<tr>
<td>Nominal</td>
<td>properties, general patterns available, some constraints</td>
</tr>
<tr>
<td>High</td>
<td>constraints, few modeling patterns, considerable number of constraints</td>
</tr>
<tr>
<td>Very High</td>
<td>instances, no patterns, considerable number of constraints</td>
</tr>
</tbody>
</table>

3.2. The ONTOCOM Cost Drivers

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, 18, 30] let us assume that one of the most significant factor is the size of the ontology. In addition, we differentiate among product, project and personnel cost drivers. Product refers to the actual ontology; the category accounts for the influence of product properties on the overall costs; that is, the characteristics of the ontology to be developed. The project category states the dimensions of the engineering process which are relevant for cost estimation; this includes, most notably, the way the project is organized and supported in terms of tools. The last category, the personnel cost drivers, emphasizes the role of team experience, ability and continuity for the effort invested in the process.

The ONTOCOM cost drivers have been defined after extensively surveying ontology engineering literature of the last two decades, and conducting expert interviews, and from empirical findings of numerous case studies in the field [26, 36]. For each cost driver we specified in detail the decision criteria which are relevant for the user of the model in order for her to determine the rating which best suits a particular project 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 decision criteria associated with a cost driver are typically more complex than

---

4See also http://ontocom.sti-innsbruck.at/
in the previous example and might be divided into further sub-categories, whose impact is aggregated to a final rating value by means of normalized weights [26]. An example is the cost driver $DCPLX$, which stands for the costs of one of the most challenging phase of an ontology development process, the analysis of the domain and the specification of the requirements the ontology is expected to fulfill. The decision which concepts will be included and in which form they will be represented in an ontology depends not only on the domain to be modeled, for instance, automotive, tourism or eBusiness, but also on the application scenario in which the ontology will be deployed. The latter includes the technical setting and the characteristics of the application in which the ontology is designed to be integrated into. As a third decision area we introduced the sources which could be eventually used as additional domain descriptions and thus as an aid for the domain analysis and the subsequent conceptualization. The global value of the $DCLPX$ driver is a weighed sum of the aforementioned areas, which are summarized in Table 2.

When using the model the project manager needs to select for each cost driver the rating level that best fits the project setting for which the estimation is carried out.

*Product-related cost drivers.* The complexity of the target ontology in ONTOCOM is described by means of several cost drivers, associated with the efforts arisen in key activities typically undertaken in ontology engineering projects. We analyzed features that are responsible for cost increases along these dimensions - independently of the size of the final ontology, the competence of the team involved and the overall project organization, which are treated separately - and aligned them to ratings from very low to very high for quantification purposes. The resulting cost drivers are

**Domain Analysis Complexity (DCPLX)** As explained in an earlier example, $DCPLX$ accounts for the efforts occurred during the domain analysis phase that initiates every ontology engineering project. The cost driver is calculated as a weighed sum of three values, which stand for the complexity of the domain itself, the comprehensiveness of the requirements analysis, and the availability of easily processable auxiliary materials. We identified characteristics of these three areas which usually influence the development efforts. For the ontology domain, we
## Domain Complexity

<table>
<thead>
<tr>
<th>Rating Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td>narrow scope, common-sense knowledge, low connectivity</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td>narrow to moderate scope, common-sense or expert knowledge, low connectivity</td>
</tr>
<tr>
<td><strong>Nominal</strong></td>
<td>moderate to wide scope, common-sense or expert knowledge, moderate connectivity</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>moderate to wide scope, common-sense or expert knowledge, high connectivity</td>
</tr>
<tr>
<td><strong>Very High</strong></td>
<td>wide scope, expert knowledge, high connectivity</td>
</tr>
</tbody>
</table>

## Requirements Complexity

<table>
<thead>
<tr>
<th>Rating Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td>few, simple requirements</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td>small number of non-conflicting requirements</td>
</tr>
<tr>
<td><strong>Nominal</strong></td>
<td>moderate number of requirements, with few conflicts, few usability requirements</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>high number of usability requirements, few conflicting requirements</td>
</tr>
<tr>
<td><strong>Very High</strong></td>
<td>very high number of requirements with a high conflicting degree, high number of usability requirements</td>
</tr>
</tbody>
</table>

## Information Sources Complexity

<table>
<thead>
<tr>
<th>Rating Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td>high number of sources in various forms</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td>competency questions and text documents available</td>
</tr>
<tr>
<td><strong>Nominal</strong></td>
<td>some text documents available</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>some unstructured information sources available</td>
</tr>
<tr>
<td><strong>Very High</strong></td>
<td>none</td>
</tr>
</tbody>
</table>

Table 2: The Domain Complexity Cost Driver $DCPLX$
considered the scope (narrow, moderate, wide), the commonality of the knowledge to be captured (common-sense vs expert knowledge), and the connectivity of the domain. The latter is expressed in the number of interdependencies between domain concepts with ranges again among three levels (low, moderate and high), while the scope is a feature which is related to the generality, but also to the perceived amount of knowledge comprised per default in a certain domain. To give just a few examples, a domain such as a department of an organization is considered narrower than the one describing a university, while the scope of the economics domain is classified as wide. The three aspects are prioritized according to common practices in the ontology engineering area, so that the connectivity of the domain is considered decisive for establishing the rating of this cost factor. The complexity of the requirements which are to be taken into consideration when developing an ontology is characterized by the total number of requirements available in conjunction with the rate of conflicting ones, and the rate of usability requirements, since the latter are seen as a fundamental source of complexity for the building process. Finally the availability of information sources guiding the engineering team during the development process or offering valuable insights in the domain to be modeled can be a major success factor in ontology engineering. When deciding upon the impact of the information sources on the effort required to perform the domain analysis activity we suggest considering the number, type and format of the sources.

Conceptualization Complexity (CCPLX) The conceptualization complexity accounts for the impact of the structure of the conceptual ontology (classes, class hierarchy, properties, instances etc) on the overall development costs. Factors decreasing these costs are support techniques such as modeling patterns. Factors that might lead to an increase in costs are the existence of specific naming and modeling constraints. This is reflected in the definition of the ratings in Table 1.

Implementation Complexity (ICPLX) One of the basic assumptions in ONTOCOM is that the most significant factor to determine the overall development costs is the size of the conceptual model underlying the ontology, expressed in entities
such as classes, attributes, relationships and instances. The actual implementation is understood to be a matter of tools, since a manual encoding of a conceptualization in a particular formal representation language is not common practice. Nevertheless, additional efforts might be required in situations in which the usage of a specific representation language is mandatory, and the encoding of the conceptual model in this language is not straightforward. In this case the implementation of the ontology requires a non-trivial mapping between the knowledge level of the conceptualization and the paradigm underlying the representation language.

**Instantiation Complexity (DATA)** The population of an ontology and the associated testing operations might be related to considerable costs. This cost driver attempts to capture the effect instance-data requirements have on the overall process. In particular the form of the instance data and the method required for its ontological formalization are significant factors for the costs of the engineering process. On the basis of a survey of ontology population and learning approaches, we assume that the population of an ontology with available instance data with an unambiguous semantics can be performed more cost-effective than the processing of relational tables or XML-structured data. Further on, the extraction of ontology instances from poorly structured sources such as free-text documents is assigned the highest value magnitude, due to the complexity of the task itself and of the pre-processing and post-processing activities. In the future we intend to revisit the scope of this cost driver, taking into account the most recent developments in the area of Linked Data. This cost driver could also serve as a starting point for an analysis of the economics of Linked Data publishing and mapping.\(^5\) Manually created instances, which may be part of the actual conceptualization of the ontology - depending on the knowledge representation language used for the implementation - are not in the scope of this cost driver,

but are covered by the Size parameter of the ONTOCOM equation.

**Evaluation Complexity (OE)** This driver stands for the effort invested in the assessing the quality of the ontology, including testing, reviewing, usability and ontological evaluation. It considers the level of activity required to test a preliminary ontology against its requirements specification document and for documentation purposes. For projects following a reuse-oriented approach we defined a dedicated cost driver which is related to activities by which the engineering team assesses the relevance and usefulness of an external ontology against the requirements of the application at hand.

**Documentation Needs (DOCU)** DOCU is a measure of the additional costs caused by detailed documentation requirements. We differentiate among five complexity levels ranging from very low (many life cycle needs uncovered) to very high (very excessive for life cycle needs).

**Required Reusability (REUSE)** Reusability is a major issue in the ontology engineering community, due to the inherent nature of ontologies, as artifacts for knowledge sharing and reuse. There is no commonly agreed understanding of the criteria required by an ontology in order to increase its reusability. Typically reusability is mentioned in the context of application independency, in the sense that it is assumed that application-dependent ontologies are likely to imply significant customization costs when reused. Additionally several types of ontologies are often presumed to encourage an increased reusability: core ontologies and upper-level ontologies describing general aspects of the world are often used in alignment tasks in order to ensure a correct ontological grounding. The Formal Ontological Analysis of Guarino [16] mentions three levels of generality, which might be associated with different reusability degrees: upper-level ontologies are used as ontological commitment for general purpose domain and task ontologies, while the latter two are combined to realize so-called application ontologies that are part of an actual information management system. These levels form the baseline for the definition of the ratings of this cost driver.
Personnel-related cost drivers. This category of cost drivers emphasizes the role of team experience, ability and continuity with respect to the effort invested in the engineering process. It contains the following four cost drivers

Ontologist/Domain Expert Capability (OCAP/DECAP) The development of an ontology requires the collaboration between a team of ontology engineers (ontologists), usually with an advanced technical background, and a team of domain experts that provide the necessary know-how in the field to be ontologically modeled. OCAP and DECAP account the perceived ability and efficiency of the overall teams of ontology specialists, and domain experts, respectively. This pair of cost drivers is defined in terms of 5 ratings defined in percentiles ranging from 15% for less capable teams up to 95% for very capable ones. The percentiles are meant to give a quantitative basis for comparison between the individual rating levels.

Ontologist/Domain Expert Experience (OEXP/DEEXP) This pair of cost drivers is complementary to the previous ones; they reflect the experience of the engineering team consisting of both ontologists and domain experts in an ontology engineering project. These drivers are not related to the abilities of single team members, but directly to their experience in constructing ontologies (OEXP), and in conceptualizing a specific domain (DEEXP), respectively.

Language/Tool Experience (LEXP/TEXP) The aim of these cost drivers is to measure the level of experience of the project team with respect to the language in which the ontology is implemented, and the ontology management tools, respectively. The development of an ontology requires the usage of knowledge representation languages with appropriate expressivity (such as Description Logics or Prolog), supported by tools such as editors, validators and reasoners. The distinction among language and tool experience is justified by the fact that while ontology languages are built on top of established knowledge representation languages from the artificial intelligence field, and are thus potentially familiar to the ontology engineer in some alternate form, tool experience implies explicitly the actual usage of a given ontology management tool, and is not directly
conditioned by the know-how of the engineering team in logics and knowledge representation.

**Personnel Continuity (PCON)** Frequent changes in the project team are a major obstacle for the success of any project within given budget and time constraints. This cost driver attempts to measure this additional overhead, where turnover rates of less than a year are assumed to lead to increased development cost in the project.

*Project-related cost drivers.* This category of cost drivers relates to general characteristics of an ontology engineering project, and their impact on the total costs. We identified three cost drivers in this category, as follows

**Support Tools for Ontology Engineering (TOOL)** We take account of the different levels of tool support for the different phases of an ontology engineering process by means of a single general-purpose cost driver, and calculate the final value as the average tool support across the entire process. The ratings for tool support are defined at a general level, comparing the degree of automation provided by the tools with the amount of manual labor required to complete a particular activity.

**Multisite Development (SITE)** This cost driver mirrors the usage of the communication support tools in a geographically distributed team, including email, phone conferences, and face-to-face meetings.

**Required Development Schedule (SCED)** This cost driver takes into account the changes induced in the engineering process by schedule constraints. Accelerated schedules tend to produce more efforts in the refinement and evolution steps due to the lack of time required by an elaborated domain analysis and conceptualization. Stretch-out schedules generate more effort in the earlier phases of the process, while the evolution and refinement tasks can be best-case neglected.

In the next section we will explain how these cost drivers are used to calculate effort estimates.
3.3. The Parametric Equation

The parametric method integrates the efforts associated with each component of the work breakdown structure introduced earlier into a mathematical formula, whose result is expressed in person-months.

\[ PM = A \cdot \text{Size}^\alpha \cdot \prod CD_i \]  \hspace{1cm} (1)

In Equation 1 the parameter \( \text{Size} \) corresponds to the size of the ontology, that is, the number of entities 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 with respect to the size of the ontology is captured by the parameter \( \alpha \). The constant \( A \) represents a baseline multiplicative calibration constant in person-months, standing for the costs which occur “if everything is normal”. The cost drivers \( CD_i \) (see Section 3.2) 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 referred to as \textit{effort multiplier} \( EM_i \). The \textit{productivity range} \( PR_i \) of a cost driver measures the ratio between the highest and the lowest effort multiplier of the cost driver \( PR_i = \frac{\max(EM_i)}{\min(EM_i)} \). It is an indicator for the relative importance of a cost driver for the overall effort estimation [1].

In order to determine the actual values of the effort multipliers and to select relevant and non-redundant cost drivers we followed a two-stage approach: first experts estimated the a-priori effort multipliers based on their experience as regarding ontology engineering. More precisely, they were asked to provide an estimate for the productivity range of each cost driver, from which the values of the rating levels have been automatically calculated so that the effort multiplier of the normal level of each driver is set to 1. Second we applied statistical techniques such as preliminary data analysis, regression and Bayes analysis to calibrate the effort multipliers in accordance to historical data from ontology engineering projects, as elaborated in the next section.
<table>
<thead>
<tr>
<th>No</th>
<th>Criterion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Definition</td>
<td>- clear definition of the estimated and the excluded costs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- clear definition of the decision criteria used to specify the cost drivers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- intuitive and non-ambiguous terms to denominate the cost drivers</td>
</tr>
<tr>
<td>2</td>
<td>Objectivity</td>
<td>- objectivity of the cost drivers and their decision criteria</td>
</tr>
<tr>
<td>3</td>
<td>Constructiveness</td>
<td>- human understandability of the model estimates</td>
</tr>
<tr>
<td>4</td>
<td>Detail</td>
<td>- accurate phase and activity breakdowns</td>
</tr>
<tr>
<td>5</td>
<td>Scope</td>
<td>- usability for a wide class of ontology engineering processes</td>
</tr>
<tr>
<td>6</td>
<td>Ease of use</td>
<td>- easily understandable inputs and options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- easily assessable cost driver ratings based on the decision criteria</td>
</tr>
<tr>
<td>7</td>
<td>Prospectiveness</td>
<td>- model applicability in early phases of the project</td>
</tr>
<tr>
<td>8</td>
<td>Stability</td>
<td>- small differences in inputs produce small differences in outputs</td>
</tr>
<tr>
<td>9</td>
<td>Parsimony</td>
<td>- lack of highly redundant cost drivers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- lack of cost drivers with no appreciable contribution to the results</td>
</tr>
<tr>
<td>10</td>
<td>Fidelity</td>
<td>- reliability of cost estimating</td>
</tr>
</tbody>
</table>

Table 3: The ONTOCOM Evaluation Framework

4. Evaluation

The applicability and usefulness of ONTOCOM have been evaluated based on the quality framework for cost models by Boehm [1], which was adapted to the particularities of ontology engineering. The framework consists of ten evaluation criteria covering a wide range of quality aspects, from the reliability of the estimates to the model ease-of-use and its relevance for arbitrary ontology engineering scenarios (see Table 3).

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, with respect to their relevance to cost issues. This part of the evaluation corresponds to criteria 1 to 8 in Table 3. Second the projections of the model were compared with data from real-world projects (criteria 9 and 10 of the quality framework). This was carried out in two iterations on two data sets containing 36 and 148 data points, respectively.
4.1. The Expert-based Evaluation

The evaluation of the a-priori model was performed by conducting interviews with two non-related groups of experts in the area of ontology engineering. Participants were given a one-hour overview of the ONTOCOM approach, followed by individual interviews that were organized according to the criteria depicted in Table 3. The feedback of the experts took into account the original scope for which the method was developed, which are ontology engineering projects with a pre-defined, or at least easily identifiable work breakdown structure, and undertaking activities according to a waterfall or spiral-like life cycle model. As briefly discussed earlier, scenarios characterized by an agile engineering approach, or in which the development of the ontology was heavily automated were not considered in the original method, and are thus not subject of the evaluation.

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 in response to the comments collected from the interviewees.

Objectivity  The objectivity of the cost drivers and the associated decision criteria were evaluated by the participants favorably. Both suffered minor modifications to accommodate the feedback received. Regarding the size of the ontology, a key parameter of the model, some interviewees expressed the need for a more careful distinction between the impact of the different types of ontological primitives (e.g., classes, attributes, relationships) with respect to the total efforts. The current version of the model does not implement this concern, while achieving reliable projection accuracy. Extensions of ONTOCOM focusing on particular species of ontology-like structures are even less affected by this issue, as the range of modeling primitives they support is limited, and does not include advanced primitives such as constraints, which are less accessible to non-experts, and hence the main reason for the concern.
**Detail/Scope** The interviewees unanimously perceived all cost drivers introduced in Section 3.2 as being relevant for ontology engineering projects. The collection of empirical data demonstrated that the method accommodates well to many real-world settings, though some ontology engineering scenarios could be surveyed only superficially due to the absence of relevant data. However, the majority of the evaluators emphasized the need of a revised model for reuse and evolution purposes, an issue which we briefly discussed earlier and that will be investigated in the future. With respect to the detail of the cost drivers, three new product drivers stating for the complexity of the domain analysis, conceptualization and implementation ($DCPLX$, $CCPLX$ and $ICPLX$, see Section 3.2) were introduced in return to an original cost driver **Ontology Complexity** ($OCPLX$). In the course of the integration of ONTOCOM with the NeOn Toolkit, more precisely with the gOnt plug-in which deals with planning and scheduling issues of ontology engineering projects [15] some of the concerns raised during the interviews have been confirmed; one of the main components of the NeOn methodology is dedicated to the various flavors in which reuse and re-engineering can be applied in the course of an ontology engineering project, suggesting alternative approaches for defining the corresponding cost drivers within ONTOCOM [37].

**Ease of use** The goal and the scope of the model were easily understood by the interviewed experts. 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, attributes, relationships/properties, rules, constraints, manually encoded instances).

**Prospectiveness** Some of the participants expressed concerns with respect to the accurate determination of the input parameters required by the method already in an early stage of the engineering process. This is acknowledged as one of the key challenges of the parametric cost estimation approach that has been addressed on many occasions in the software engineering literature [1, 18]. A common assumption is that the reliability of the input data will increase as project managers gain more experience with ontology engineering projects and can assess
the value of the parameters by comparison with previous project situations they have been involved in.

**Stability** This is ensured by the mathematical model underlying ONTOCOM.

The remaining two evaluation criteria **Fidelity** and **Parsimony** were approached through the statistical calibration of the ONTOCOM model.

### 4.2. First Evaluation of the Estimation Quality

#### 4.2.1. Data Collection

The first iteration of the calibration was based on the data collected from 36 structured interviews with ontology engineering experts. The interviews were conducted within a three months period and covered 35 pre-defined questions related to the cost drivers listed in Section 3.2. The survey participants were representative for the community of users and developers of semantic technologies. The group consisted of individuals affiliated to industry and academia, who were involved in the last 3 to 4 years in ontology development projects in different domains.

#### 4.2.2. Calibration Method

In order to adapt the 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 [31] combined with Bayes analysis [3]. This approach allows the concomitant 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. According to Bayes’ theorem, initial values of the effort multipliers and their rating levels are used to calculate a posterior distribution of the parameters used in the ONTOCOM equation based on the variances of the prior, expert-defined values, and the samples available as historical data. If the variance of the prior information is smaller

---

6A detailed presentation of the structure of the questionnaire and a discussion of the results can be found in [27]. The survey is available online at [http://survey.sti2.at/public/survey.php?name=OntocomSurveyJune13](http://survey.sti2.at/public/survey.php?name=OntocomSurveyJune13).
than the variance observed in the data set used for the calibration, the former is considered more reliable and assigned a higher weight. Otherwise, the values calculated using multilinear regression are weighed higher, causing the a-posteriori values of the parameters to be closer to the empirical data.

7 The analysis resulted in the exclusion of some of the cost drivers to increase the accuracy of the results.

4.2.3. Calibration Results

The approximation of the effort multipliers via linear regression implied a reformulation of the ONTOCOM equation by introducing scaling factors by which the existing effort multipliers are scaled in order to fit the model. The adapted linear regression delivers a best-fit for the effort multipliers with respect to the surveyed empirical data. However, the relatively small sample size results in a limited accuracy of the estimated value. This drawback can be overcome with the help of the a-priori estimations of the parameters defined by human experts, which are combined with the empirically determined factors using Bayes analysis.

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

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

7Refer to [10] for an exhaustive explanation of the application of Bayes analysis for cost estimation purposes.
Table 4 summarizes the results of the Bayes 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 yielded positive results, for instance, for the exponent $\alpha$ (*SIZE*), but it was less positive for the effort multipliers related to *OCAP/DECAP*. The *Productivity range* column lists the relative influence a cost driver has on the final estimate. It is defined as the ratio between the highest and the lowest value assigned to the rating levels of the corresponding cost driver. The higher this value, the stronger the impact of the cost driver on the total costs.

Figures 3 and 4.2.3 compare the outcome produced by the calibrated model with the collected data. In order to visualize the results we normalized the data with the product of the corresponding cost drivers. The gray lines indicate a range around the projections adding or subtracting 75% of the effort delivered by ONTOCOM. This result can be interpreted as follows: 75% of the historical data points we have lie within this range, calculated by modifying the estimate up or down with 75%. For a narrower range of 30%, which corresponds to a higher accuracy of the projections, the model covers only 32% of the real-world data, which means that for most of the cases, this version of
ONTOCOM manages to deliver only rough estimates. Nevertheless, these results also give evidence for a linear behavior of deviation, a promising result for a first calibration of the model. The accuracy of the model has considerably improved in the second iteration of the calibration, which we elaborate upon in the next section.

4.3. Second Evaluation of the Estimation Quality

4.3.1. Data Collection

The second calibration of ONTOCOM started with an extensive survey, in which participants in ontology development projects were interviewed with respect to aspects related to the different cost drivers of the model. The survey was supported by the same online questionnaire as in the first iteration and contained data from 148 ontology development projects. Experiences in effort estimation in related engineering disciplines, notably software engineering, let us assume that this data set is large enough to form the basis for a reliable cost model of the size of ONTOCOM [9].

Approximately 50% of the data was collected during face-to-face or telephone interviews, the rest via the self-administered online questionnaire. Approximately 95% of the ontologies collected were built in Europe, whilst nearly 35% originated from industry parties. The size of the ontologies in the data set varied from 60 to 11 million entities. Most ontologies were implemented in OWL DL (approximately 30%), followed by WSML DL and WSML Flight (around 10% each) and RDF(S) (9%). The duration of the projects ranged from 0.02 to 156 person-months.

4.3.2. Calibration Method

We then calibrated the model on this enlarged data set using the same statistical methods as in the previous iteration. This led to a comparatively minor improvement in the estimation accuracy. The previous calibration of the model, on 36 data points, resulted in an accuracy of 31.82% within a ±30% error margin. The new calibration, based on a three-times larger data set, led to an accuracy improvement of around 2% in the same margin. The reason for this minimal change could be traced back to the

---

8A detailed discussion of the results can be found in [36].
issue of unbalanced data sets [4]. To solve this issue, and to enable the model to reach an accuracy level which is compelling to real-world enterprise settings, we needed to further analyze the nature of the data collected, as described for instance in [19, 20, 22, 23].

As a next step we conducted a preliminary data analysis on the enlarged data set following the framework of Lin & Mintram [22]. We were particularly interested in the first six steps of the framework, which dealt with the identification and removal of outliers from data sets and with the determination of the cost drivers with the greatest impact on the estimated effort. The preliminary data analysis resulted in a considerable improvement of our results. First, outliers in the numerical variables were removed, and the remaining data was transformed to a normal distribution. The same was applied to ordinal variables and the entire process was repeated until all outliers were eliminated. Once this was achieved, we performed a correlation and regression analysis, followed by an analysis of variance (ANOVA) on the ordinal variables. Finally, the data set was calibrated (using multivariate regression and Bayes analysis) in order to combine the empirical data with the values of the effort multipliers provided by human experts in the a-priori model.

The ONTOCOM model contains numerical, as well as ordinal variables. There are two numerical variables, $Person - Months (PM)$, which is a response variable, and $Size$, which is an exploratory variable. For the calibration of the model we considered only those cost drivers for which expert data was available (see Table 4.3.2). The ratings collected for each cost driver for each data point (ranging from very low to very high) were transformed to nominal numbers between 1 and 5.

In the following we briefly describe each step of the framework, and discuss the results of the subsequent calibration.

**Step 1: Outlier Detection in Numerical Variables.** Table 4.3.2 summarizes the analysis of the numerical variables $PM$ and $Size$. For each variable the name of the variable,
Table 5: Variables Used in the Preliminary Data Analysis

<table>
<thead>
<tr>
<th>Cost Driver</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM</td>
<td>Person-months</td>
<td>Numerical</td>
</tr>
<tr>
<td>SIZE</td>
<td>Size of the ontology</td>
<td>Numerical</td>
</tr>
<tr>
<td>DCPLX</td>
<td>Domain Complexity</td>
<td>Ordinal</td>
</tr>
<tr>
<td>CCPLX</td>
<td>Conceptualization Complexity</td>
<td>Ordinal</td>
</tr>
<tr>
<td>ICPLX</td>
<td>Implementation Complexity</td>
<td>Ordinal</td>
</tr>
<tr>
<td>REUSE</td>
<td>Required Reusability</td>
<td>Ordinal</td>
</tr>
<tr>
<td>DOCU</td>
<td>Documentation Needs</td>
<td>Ordinal</td>
</tr>
<tr>
<td>OE</td>
<td>Ontology Evaluation</td>
<td>Ordinal</td>
</tr>
<tr>
<td>OCAP/DECAP</td>
<td>Ontologist/Domain Expert Capability</td>
<td>Ordinal</td>
</tr>
<tr>
<td>OEXP/DEEXP</td>
<td>Ontologist/Domain Expert Experience</td>
<td>Ordinal</td>
</tr>
<tr>
<td>PCON</td>
<td>Personnel Continuity</td>
<td>Ordinal</td>
</tr>
<tr>
<td>LEXP/TEXP</td>
<td>Language/Tool Experience</td>
<td>Ordinal</td>
</tr>
<tr>
<td>SITE</td>
<td>Multisite Development</td>
<td>Ordinal</td>
</tr>
</tbody>
</table>

the number of data points, the range of values and the mean and standard deviation are shown. The variable $PM$ ranges from 0.02 to 156 person-months, with an average size of 10.22 person-months at a standard deviation of 25.60. $Size$ values are between 0.06 and 11000 kilo entities, with an average size of 76.14 kilo entities at a standard deviation of 904.09.

Table 6: ONTOCOM Numerical Variables

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Data points</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM</td>
<td>148</td>
<td>0.02</td>
<td>156</td>
<td>10.22</td>
<td>25.60</td>
</tr>
<tr>
<td>Size</td>
<td>148</td>
<td>0.006</td>
<td>11000</td>
<td>76.14</td>
<td>904.09</td>
</tr>
</tbody>
</table>

The outlier detection was performed using box plots, which offer a graphical means of depicting groups of numerical data. Our initial run identified 5 data points as outliers that were removed from the subsequent analysis.

Step 2: Data Transformation. In this step we transformed the data so that it approximates a normal distribution. As both numerical variables were not normally distributed,
we created new variables, named $LnPM$ and $LnSize$, by applying the natural logarithm, and tested their distribution using histograms, as shown in Figures 4 and 5. The new logarithmic variables were used in the subsequent steps of the preliminary data analysis.

![Figure 4: Histogram of the Variable LnPM](image1)

![Figure 5: Histogram of Variable LnSize](image2)

**Step 3: Outlier Detection in Ordinal Variables.** Ordinal variables correspond to the three categories of cost drivers in the ONTOCOM model introduced in Section 3.2. The outlier analysis was performed again using box plot diagrams and led to the removal of 44 data points (in two iterations). The box plot in Figure 6 depicts the remaining data set that is free of outliers in the ordinal variables. We also re-measured the normal distribution of this data set through Quantile-Quantile (Q-Q) normality plots for the numerical variables $LnPM$ and $LnSize$ that were created in Step 2 (see Figures 7 and 8).

Steps 4 to 6 of the preliminary data analysis and the subsequent calibration were performed on the remaining data set of 99 ontology development projects - 5 data points being eliminated after the detection of outliers in the numerical variables, while other 44 data points were removed during the same procedure applied to the ordinal variables of the model.

**Steps 4 and 5: Correlation Analysis and Regression Analysis.** Correlation analysis involves calculating a correlation coefficient that measures the relationship between the response and the explanatory variable. The coefficient has a value between $+1$ and $-1$, 

29
where a value greater than 0 indicates positively correlated variables, in other words, a rise of the explanatory variable will result in a rise of the response variable, whilst a decrease in the explanatory variable will result in a decrease of the response variable. The value is an indicator on how strong this relationship is, with 1 staying for a pair
of perfectly positively correlated variables. Conversely, if the value is below 0 then an increase in the explanatory variable will result in a decrease of the response variable and vice versa, with −1 standing for variables that are perfectly negatively correlated. 0 means that the variables are not correlated. As measures for the correlation analysis we used the Spearman’s rank coefficient (for ordinal variables) and Pearson’s rank coefficient (for interval or ratio type data), where the threshold values were set to $Pr > |0.65|$ and $Pr < |0.1|$, for high and low correlation, respectively. The correlation analysis indicated no correlation in the numerical or ordinal variables. A subsequent regression analysis confirmed that the correlation between $LnSize$ and $LnPM$ is within the allowed intervals.

**Step 6: Stepwise ANOVA Analysis.** In the next step we performed a stepwise ANOVA (analysis of variance) analysis to compute the best 6-variable ONTOCOM model. ANOVA is a statistical method commonly used to analyze data sets with ordinal type variables [4, 19, 23, 22]. The data set variables are referred to as factors, and the scale values of the variables as factor levels. ANOVA tests whether there are significant differences between factor level means. It is assumed that the variance of the level means will be large if there are differences between the groups, and small if there are none. The variance of the means between groups is divided by the variance within the groups to calculate the F-statistic measure (see Equation 2). The result is close to 1 if the group means are not significantly different from one another.

$$F = \frac{\text{Variance between}}{\text{Variance within}}$$  \hspace{1cm} (2)

The best 6-variable model is defined iteratively. In each iteration one identifies the next explanatory variable that has the most impact on the response variable. This is achieved by examining the adjusted coefficient of determination (i.e., the $R^2$ value). The variable with the highest adjusted $R^2$ is the variable that best explains the variation in the response variable, and is thus kept in the model [22]. In each iteration the adjusted $R^2$ values are re-calculated in order to account for the variables previously selected.
Applying this procedure to ONTOCOM and our data set revealed that the six variables that best explain the behavior of the effort are (in this order): Domain Complexity \((DCPLX)\), Ontology Evaluation \((OE)\), Language/Tool Experience \((LEXP/TEXP)\), Ontologist/Domain Expert Capability \((OCAP/DECAP)\), Documentation needs \((DOCU)\), and Personnel Continuity \((PCON)\). Our analysis based on the adjusted \(R^2\) values also pointed out that any additional variables added to the model would not have a significant impact on the response variable.

4.3.3. Calibration Results

The preliminary data analysis resulted in a set of 99 data points and identified the six variables which best explain the variation in the response variable. We performed a calibration on this best 6-variable model to combine the expert opinion with the data collected. Our results showed a substantial improvement of the accuracy of estimating results within a \(\pm 30\%\) margin from 33.78\% to 45.82\% and within a \(\pm 75\%\) margin from 66.89\% to 79.80\%. We then performed our calibration with all 11 variables to examine the impact of the other variables. The result was a slight increase in accuracy of 46.46\% within the \(\pm 30\%\) error margin, whilst for the \(\pm 75\%\) margin the differences were even less significant.

The analysis of the behavior of the cost drivers is consistent with the findings from the first calibration which was based on a significantly smaller data point set, particularly for two of the three most dominant cost drivers, Domain Complexity \((DCPLX)\) and Ontology Evaluation \((OE)\). Additionally, we examined the behavior of productivity, i.e., the ratio between the \(Size\) and \(PM\) variables, and accuracy for the larger margin error of \(\pm 75\%\). The box plot of the productivity factor showed a stable set with only 7 outliers from the full data set. The accuracy for delivering results within \(\pm 75\%\) increased from 66.67\% to 79.80\% after the preliminary data analysis. This supports the fact that ontology development projects rarely exhibit any erratic behavior in their costs, in other words, there are only rare cases when the actual costs might exceed the anticipated costs by several times.
5. Using ONTOCOM

In accordance to the work breakdown structure introduced earlier the estimation of the costs can be performed during the feasibility study or as part of the requirements analysis in the course of an ontology engineering project. The second option is likely to lead to more reliable projections due to the fact that many of the aspects which are reflected in cost drivers of the ONTOCOM model are expected to be intensively studied during the requirements analysis phase, thus making accurate approximations of the input parameters possible: the expected size of the ontology, the engineering team, the tools to be used, the knowledge representation language in which the ontology will be implemented, and so on.

Following ontology engineering methodologies such as Methontology[12, 14] and OTK [40] - to which ONTOCOM was targeted per design - the requirements phase of every engineering project includes an analysis of the prospected size of the ontological artifact to be developed. This analysis takes into account previous experience in similar types of projects (e.g., ontology-based recommender systems, ontology-based search, essentially considering the type of technical system for which the ontology will be used) and vertical domains. The size of the ontology is measured in thousands of entities, and for the calculation of cost projections an estimate of the expected order of magnitude of the ontology is likely to be sufficient to achieve reliable accuracy levels. Consulting related projects and case study in the literature may provide insights with respect to the typical scope and size of an ontology in a given setting; for instance, knowledge representation projects in the life sciences domain attest the fact that ontologies in this area tend to have thousands to tens of thousands of entities, a figure which can be taken as a baseline for the assessment of the current project setting. Translating the number of entities to the actual implementation depends on the differences between the expressivity of the conceptualization and the paradigms beyond the used representation language.

The project manager then assesses the suitable rating level of each of the cost drivers applicable to the project at hand, in accordance to the information available at this point. Depending on their impact on the overall development effort, if a partic-
ular activity increases the nominal efforts, then it should be rated with values such as high and very high. Otherwise, if it is perceived to cause a decrease of the nominal costs, then it should be rated with values such as low and very low. Cost drivers which are not relevant for a particular scenario, or are expected to have a nominal impact on the overall estimate, should be set to 'normal', which corresponds to the value 1 and thus does not influence the result of the model. The first step is the estimation of the size of the ontology to be developed, expressed in thousands of entities: if we consider, say, an ontology with 1000 classes, 200 relationships (including sub-class-of) and 100 rules, the size parameter of the estimation formula will be calculated as follows:

\[ \text{Size} = \frac{1000 + 200 + 100}{1000} = 1.3 \]  

(3)

Assuming that the ratings of the cost drivers are those depicted in Table 7 these ratings are replaced by numerical values calculated through expert judgement (in the a-priori model) and statistical calibration (in the a-posteriori model). For the particular case illustrated in Table 7 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. According to the formula 1 \( A = 2.92 \) the total development effort of 11.44 PM results from the following:

\[ PM = 2.92 \times 1.3^{1} \times (1.26 \times 1^{10} \times 1.15 \times 1.11 \times 0.93 \times 1.11 \times 0.89 \times 1.2 \times 1.7) \]  

(4)

The value of the parameter \( A \) has been determined through the statistical calibration of the model, while economies of scale \( \alpha \) are so far not taken into consideration.

In order to use ONTOCOM in a particular setting - characterized by a given enterprise, business domain, certain types of ontologies, or an enterprise-specific engineering methodology, to name only a few possible scenarios - the generic method can be adjusted to reflect this additional knowledge in the associated cost drivers and the way they are interconnected via the cost model. Based on our experiences so far we recommend the following basic steps to achieve this
### Table 7: Example of ONTOCOM Use

<table>
<thead>
<tr>
<th>Cost driver</th>
<th>Effort</th>
<th>Value</th>
<th>Cost driver</th>
<th>Effort</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product factors</strong></td>
<td></td>
<td></td>
<td><strong>Personnel factors</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCPLX</td>
<td>High</td>
<td>1.26</td>
<td>OCAP</td>
<td>High</td>
<td>1.11</td>
</tr>
<tr>
<td>CCPLX</td>
<td>Nominal</td>
<td>1</td>
<td>DCAP</td>
<td>Low</td>
<td>0.93</td>
</tr>
<tr>
<td>ICPLX</td>
<td>Low</td>
<td>1.15</td>
<td>OEXP</td>
<td>High</td>
<td>1.11</td>
</tr>
<tr>
<td>DATA</td>
<td>High</td>
<td>1</td>
<td>DEEXP</td>
<td>Very low</td>
<td>0.89</td>
</tr>
<tr>
<td>REUSE</td>
<td>Nominal</td>
<td>1</td>
<td>LEXP</td>
<td>Nominal</td>
<td>1</td>
</tr>
<tr>
<td>DOCU</td>
<td>Low</td>
<td>1</td>
<td>TEXP</td>
<td>Nominal</td>
<td>1</td>
</tr>
<tr>
<td>OE</td>
<td>Nominal</td>
<td>1</td>
<td>PCON</td>
<td>Very high</td>
<td>1</td>
</tr>
<tr>
<td><strong>Project factors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOOL</td>
<td>Very Low</td>
<td>1</td>
<td>SITE</td>
<td>Nominal</td>
<td>1</td>
</tr>
</tbody>
</table>

- 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 parameterized statistical model

- Calibrate the a-priori model based on previous project data to create a more accurate a-posteriori model

- Use the calibrated model to calculate the expected development costs.

In its most generic form ONTOCOM does not consider alternative engineering strategies such as rapid prototyping, which follow an iterative life cycle. This limitation has been discussed in previous work of ours, where we described how ONTOCOM could be adjusted to suit such scenarios, using the collaborative ontology engineering methodology DILIGENT as an example [26, 28]. In particular, the model does not explicitly support activities to ontology maintenance, including the evolution of the instance base which are incrementally added to the ontology. To estimate this type of

---

10 Refer to [14] for a discussion on the relation between this process model and the IEEE standards [17].
costs it is recommended that the engineering team defines specific time points and milestones related to releases of the ontology and re-assesses the values of the parameters of the ONTOCOM equation for the share of the ontology which is expected to be built in the next iteration. An alternative estimation approach amenable to such scenarios is addressed in other work of ours within the context of FOLCOM [35]. In addition to these design-level considerations, in practice ONTOCOM may be less suited for ontology engineering projects that are primarily reuse-oriented - by customizing and integrating existing ontologies, by (automatically) translating related knowledge structures such as thesauri, taxonomies and classifications into ontologies, or both. These scenarios are supported by the underlying estimation model, but in a non-calibrated form due to the absence of a critical mass of historical data about such ontology engineering projects; hence, we can not make any reliable statements about the quality of the projections for projects heavily relying on existing ontologies.

We developed a series of tools that assist project managers in following the steps just mentioned. The main tool is based on Microsoft Excel and provides an easy-to-use user interface for running calibrations and calculating effort estimates. The calibration can use existing data, self-owned data, or a combination of both. Furthermore, users are able to customize the method, selecting the cost drivers to be taken into account for a calibration from the available list and defining additional ones in response to the characteristics of their ontology engineering environment. In a nutshell the Excel tool consists of three pre-defined spreadsheets where users can fill in information, called Data Entry, Expert Data, and Empirical Data, respectively, and additional ones that are generated automatically in the course of a new calibration of the ONTOCOM model. The Data Entry spreadsheet depicted in Figure 9 provides an overview of the cost drivers currently used, the data included in the a-posteriori model, and the expected accuracy. The user can start a new calibration of the underlying model, or simply use it to calculate effort estimates.

The second spreadsheet Expert Data shows the expert data available for each cost driver and, in the subsequent columns, the updated productivity range and effort mul-

---

11 The tools are available online at http://ontocom.sti-innsbruck.at/tools.htm.
tipliers based on the calibration of the model. The user is allowed to change the expert values to better reflect her own opinion. The calibrated values of the cost drivers are used for effort projections. On the third spreadsheet *Empirical Data* the user has the option to add or remove data points to the calibration data set.

When the user starts a new calibration the tool creates additional tables, reflecting the individual calibration steps performed, which are highlighted in a different color. In the spreadsheet *Empirical Data (Mapping)* categorical values of the empirical data are mapped to their corresponding values from the expert data. If the latter is not available for a cost driver, the cost driver will not be used in subsequent calibration steps (see Section 4). In the *Data Entry* form (see Figure 9) the user is then notified which cost drivers were not used in the calibration due to this unavailability. In the next spreadsheet (*LogData*) the data is mapped to logarithmic values, a procedure that is required when using multivariate regression analysis. The two next spreadsheets present the results of the correlation analysis; *Correlation Table* contains information about the
correlations between explanatory and predictor variables and between the explanatory variables themselves. *Post-correlation Data* contains the selected (logarithmic) data from the previous spreadsheet based on the result of the correlation analysis. The next spreadsheet is dedicated to the *Statistical Analysis*, showing the results of the multivariate regression and Bayes analysis. For each cost driver, the result is an exponent which is used on the original values to determine a new calibrated value for each of its five rating levels. In the last spreadsheet titled *Predictions* the data points are tested against the predictions which are derived by the model using the newly calibrated values. The results are evaluated for different thresholds, which are presented as an accuracy percentage in the *Data Entry*. A second way to interact with the tool is the actual usage of the method to calculate effort estimates for a given project setting, characterized through appropriate rating levels of the cost drivers, which need to be specified by the project manager (see Figure 10).

![Figure 10: Specifying Cost Driver Values to Calculate Effort Estimates](image)

The same functionality is available as Web-based tool, operating on the generic version of the ONTOCOM model calibrated on 148 data points as explained in Section 4. A screenshot of this second tool, which targets potential users of the cost estimation method who are not specifically interested in the details of the model or customizing it for new project scenarios, is provided in Figure 11.

38
6. Related Work

Cost estimation has a long-standing tradition in more mature engineering disciplines such as software engineering or industrial production [1, 18, 39]. Although the importance of cost issues is well-acknowledged in the semantic technologies community, as to the best of our knowledge, no cost estimation model for ontology development besides ONTOCOM has been proposed so far. Analogue models for the development of knowledge-based systems (e.g., [11]) implicitly assume the availability of the underlying conceptual structures. [24] provides a qualitative analysis of the costs and benefits of ontology usage in application systems, and discusses a number of projects in which quantitative evidence of the benefits of using an ontology-based approach is available, but does not offer any model to estimate the efforts. [7] presents empirical results for quantifying ontology reuse. [6] proposes a model for analyzing and assessing the benefits of using ontologies in a given setting, but the model proposed

![Figure 11: Interface of the Web-based Tool](image)

<table>
<thead>
<tr>
<th>Ontology Size</th>
<th>100</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>The product category accounts for the influence of product properties on the overall costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dicpix</td>
</tr>
<tr>
<td></td>
<td>lchpx</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reuse</th>
<th>The reuse category accounts for the influence of reuse costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ooo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Personnel</th>
<th>The personnel category accounts for the influence of personnel experience on the overall costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ocap/decap</td>
</tr>
<tr>
<td></td>
<td>pcon</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>The project category states the dimensions of the engineering process which are relevant for the cost estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tool</td>
</tr>
</tbody>
</table>

Estimate Costs Estimated Benefit: 2.9579159054260605

![Ontologist/Domain Expert Experience](image)
is not quantitative. [43] present a cost/benefit analysis model for personal knowledge management which gives consideration to the externalization of knowledge in the form of formal statements. They apply their model to semantic wikis, but do not provide any empirical evaluation. [21] adjusts a cost estimation model for Web applications to accommodate the additional efforts induced by the adoption of ontology-based technology. The resulting cost drivers are, however, not adapted to the requirements of ontology development and no evaluation is provided. Nevertheless, the question of cost estimation for ontology-based software applications is relevant, and our framework proposed an initial model covering this aspect [5]. Wolff et al. address the efforts related to the engineering of Semantic Web Services in [44]. They argue that Semantic Web Services projects require less time compared to their traditional Web services equivalents, but do not treat ontologies, or any other semantic technology, as a separate part of their observations. Other researchers investigate related management issues pertaining to the development of ontology-based applications. [15] and [41] describe gOntt, a planning tool for ontology-based applications using Gantt charts in which activities related to the life cycle of networked ontologies are mapped to existing life cycle models. These activities can be augmented with effort-related information, as we have elaborated in [37]. In an nutshell, the project manager is expected to provide a percentage distribution of the effort across the project phases defined in the scheduling tool, and ONTOCOM estimates, which span the overall duration of the project, are distributed accordingly to each phase. In [13] the authors propose a methodology to study and assess the risks of ontology engineering, which may form a core component of the feasibility study phase of an ontology engineering project (see Section 3). An interesting line of work could be to align our method with the risk analysis methodology proposed by Ferreira et al., in a similar exercise to the one we carried out for the NeOn methodology [37]. This would lead to an improved integration of economics-motivated aspects into ontology engineering practice. An important, emerging field of research investigates mechanisms to incentivize data owners and other parties to lift data into RDF, expose it online and connect it to other data sets, and potential business models for data publishing and processing. Instruments to study the costs and benefits of the underlying activities are a prerequisite for such considerations in an enterprise con-
text; the current line of argument within the Linked Data community advertises the ‘pay-as-you-go data integration’ concept, according to which the overall effort of creating a new representation for existing data as well as links to external data sources is split along the data management life cycle among various parties, applies well to open environments such as the Web, and assumes a critical mass of participation to reduce individual costs up to an acceptable margin - similarly to early approaches to ontology engineering costs which were expected to be negligible due to the in theory highly collaborative nature of the process.

7. Conclusions

The adoption of ontology-based technologies in commercial settings depends on the availability of appropriate methodologies and software assisting the ontology engineering process, and on methods for an effective cost/benefit management. We proposed a parametric cost estimation model for ontologies which we calibrated using various statistical techniques, reaching in a second iteration an accuracy of 45.82% within a ±30% margin, and 79.80% within a ±75% margin. The data on which the calibration was based was collected from ontology engineering projects from various domains, covering ontologies built for different purposes and used in different environments. Experiences with cost estimation models suggest that a customized model, optimized for particular project or enterprise settings, would yield even better estimates [9]. This potential mismatch between the generality of the model and its suitability for specific environments can be (partially) compensated by a large, representative data set used to calibrate the model. The data set in the second calibration enabled us to develop a second release of ONTOCOM with a significantly improved estimation quality. This can be traced back to the preliminary data analysis performed, but also to the size of the data set, which is sufficiently high for the number of variables in the model. The results we obtained on the calibration on 148 data points suggest, nevertheless, that the projections of ONTOCOM as a generic model should improve with the availability of additional data. Data analysis on an extended set might show slight differences with respect to the dominant behavior of certain variables, however, our experience shows
us that the relevancy of the cost factors is very much consistent. The quality of the ONTOCOM calculations could be further optimized by establishing variants of the model tailored to reflect particularities of certain classes of ontological structures. Following this line of reasoning, we developed a number of extensions of ONTOCOM which are described in [5].

To further improve the usefulness of ONTOCOM we implemented software tools which offer on-the-fly cost estimation for ontology projects and automatic calibration support. Future tool development plans include integrating cost estimation into ontology development and management environments such as gOnt in which the user can assess costs in the same platform in which the overall ontology development process is managed.

As noted earlier, the calibration of the ONTOCOM model was based on a data set that did not sufficiently covered reuse-oriented projects. As such, while the model covers by design aspects and activities relevant for ontology reuse, it may be less adequate to deliver accurate projections for such situations due to the absence of a critical mass of historical data. Projects focusing on reuse raise additional challenges. First, there is the question of how the target ontology is obtained from the reused resources, in particular, what is the manual effort required to execute the reuse activities mentioned earlier. The answer to this question is likely to change in the light of several ICT trends we are recently witnessing; as more and more structured data is becoming openly available, and the associated data processing and management technology is maturing, a reuse-driven knowledge engineering approach is becoming not only technically, but also economically feasible. From a cost estimation perspective, this development will require new cost estimation approach, which are based on a closer analysis of the impact of heavily automatized methods and techniques to create ontologies on development costs; ONTOCOM in its current version is based on empirical data from projects where automation had played only a minor role in the overall engineering process. A second aspect that remains to be investigated is the suitability of an alternative design of a reuse-oriented cost model, similar to the extensions proposed in the context of the COCOMO model in software engineering [2]. There reuse-related activities are not considered within the work breakdown structure that is laid out to define the cost
drivers; in return, the equation uses a dedicated reuse parameter that is dependent on the share of the software artifact that is integrated in the target system with or without modifications. An analysis of the feasibility of a reuse-oriented approach in ontology engineering projects from a cost perspective is provided in [34].

In a future release of ONTOCOM we plan to revisit these considerations in order to response to insights gained from reuse practices for ontologies in the Linked Data publishing community, which provides anecdotal evidence for the reuse of simple vocabularies such as FOAF, Dublin Core and SKOS;\textsuperscript{12} though an in-depth understanding of the overall reuse phenomenon and the factors that lead to a given ontology to become more popular than others is largely missing.

Additional adjustments are needed in order to effectively support scenarios in which the ontology is created mainly with the help of automatic techniques. With the number of useful ontologies and RDF data sets steadily growing, ontology reuse becomes feasible from a technical and an economic perspective. Such a revision would nevertheless have to give consideration to the data-driven techniques for ontology engineering and learning which are emerging within the Linked Data initiative, which only loosely follow systematic procedures or methodologies as they have been classically conceived by the research community in the past. In a different work of ours we have investigated cost estimation methods which explicitly target agile development scenarios [35]; similar considerations could be the basis for the design of a cost estimation method that is fundamentally different than the parametric approach followed by ONTOCOM. The prospected method would accommodate scenarios which do not comply to process-oriented methodologies, but pursue an ontology engineering strategy combining reuse and automatic techniques (for instance, for information extraction and relational data lifting), and would also be amenable to project costs related to ontology maintenance and evolution.

Acknowledgements. The research leading to this paper was partially supported by the European Commission under the contract FP7-215040 ACTIVE.

References


