



University of Innsbruck

STI IBK
(Semantic Technologies Institute)

Integrating ONTOCOM to gOntt

Technical report

**Dr. Elena Simperl,
Dr. Katharina Siorpaes,
Prof.Dr. Sung-Kook Han,
MSc.Bakk.techn. Stephan Wölger**

Innsbruck
November 23, 2010

Contents

- Introduction 3
- NeOn, gOntt and ONTOCOM II 3
 - NeOn 3
 - Life cycle models 4
 - gOntt 5
 - Scheduling of ontology projects 5
 - Ontology project execution support 5
- ONTOCOM/ONTOCOM II 6
 - ONTOCOM 6
 - ONTOCOM II 7
- Alignment of cost drivers to the NeOn glossary 8
 - Product-related cost drivers 8
 - Cost drivers for ontology building 8
 - Cost drivers for reuse and maintenance 9
 - Personnel-related cost drivers 10
 - Project-related cost drivers 10
 - New cost drivers for ONTOCOM 10
- Applying ONTOCOM to the NeOn life cycle models 11
- How to benefit from ONTOCOM estimations 13
- Implementation mock-up 14

Introduction

NeOn is a framework for modeling ontologies for the use in large-scale semantic applications in distributed organisations. The main purposes of this toolkit are to enable collaborative engineering of ontologies as well as to guarantee dynamic and evolvable ontologies. To enrich this framework with cost estimation capabilities we want to integrate ONTOCOM II, which provides a tool to estimate the efforts involved in building, reusing and maintaining ontologies, into gOntt, a plug-in for NeOn, which provides additional functionalities focusing on project scheduling and supporting the ontology project execution.

The integration of ONTOCOM II with gOntt consists of the following steps: first, we map ontology engineering activities covered by the NeOn methodology with the cost drivers defined in the cost model. This also includes reorganisation of the ONTOCOM II model in order to provide support for new aspects arising from the NeOn glossary; then we show how cost estimation fits into the life cycle models they identified; and finally provide tool support for this integration through a (planned) extension of the NeOn plug-in gOntt, which provides additional functionalities focusing on project scheduling and supporting the ontology project execution.

Extending gOntt into cost information has to take into account several key aspects related to the parametric approach adopted by ONTOCOM II (and related models in other areas) to predict development efforts. First, such models require a critical mass of empirical data for calibration purposes, which inherently reduces the number of ontology engineering activities covered by NeOn for which accurate estimates can be calculated. Second, refining the cost model to support a higher-level of granularity would ask for the definition of various sub-models covering a specific, complex ontology engineering activity; these models, provided the availability of historical project data, could be then applied to produce estimates for specific phases of the life cycle model followed, however, they would be rather focused and restricted in application, and, for certain activities (such as domain analysis) would very likely require deeper insights into the practices that are actually in place at the enterprise using the corresponding model. Therefore, we opted for a slightly different strategy, which overcomes these issues. In a nutshell, ONTOCOM II (in its altered version resulting from the alignment between the NeOn methodology and its current cost drivers) can be used to predict the total cost of an ontology engineering project; the project manager subsequently devises a distribution function by which this estimate is broken down to the individual phases of the project, based on her expertise and on existing insights from case studies in ontology engineering literature. The simplest form of this distribution function is based on percentages, which are then taken into account to calculate cost estimates per phase or even activity.

NeOn, gOntt and ONTOCOM II

NeOn

Neon is a project which was co-funded by the European Commission involving 14 partners from Europe. NeOn is a framework for modeling ontologies for the use in large-scale semantic applications in

distributed organisations. The main purposes of this toolkit are to enable collaborative engineering of ontologies as well as to guarantee dynamic and evolvable ontologies. NeOn is based on the metamodel of Eclipse which makes it easy to modularly increase its capabilities by adding new plug-ins.

Life cycle models

In order to structure an ontology project it is convenient to use some kind of guidance in form of a methodology regarding a life cycle model. NeOn provides two different models. One is the well known waterfall model, the other one is the iterative-incremental model. Each of them has advantages and disadvantages, however both are of value depending on the specific needs of a user scheduling an ontology project.

The waterfall model is a model in which the user has to know any important aspect of the project. There must not be any ambiguities regarding requirements and there must not be any changes. At the beginning of the project the different phases get related to each other in a sequential manner which implies that one phase has to be carried out and completed before the following one can be started. Except of the maintenance phase it is not possible to go back to prior phases to add/alter/delete processes or activities. This on the one hand ensures ease of understanding however on the other hand it limits the user's flexibility. Therefore this model is used rather for modeling well known domains or for small projects. Furthermore the use of this model can be convenient when having a specific set of requirements which can be fixed in advance. The waterfall model consists of certain phases. In the simplest fashion it consists of the *initiation phase*, the *design phase*, the *implementation phase* and the *maintenance phase*. Depending on specific needs this basic model can be advanced by including additional phases like the reuse phase, the merging phase, the reengineering phase or the *modeling phase*.

Since the ONTOCOM model is about supporting the development of an ontology by providing estimations of the efforts that may arise during an ontology building project, these estimations could be used to set up a detailed time line in the waterfall model. ONTOCOM's cost drivers could be mapped to activities and processes (which are to carry out during a phase) and based on that ONTOCOM's estimations could be used as for associating personnel to phases (or finer granular levels) and thereby getting the time needed for completing a phase.

If there is the need for a more agile/flexible approach then one should use the iterative-incremental approach. This model consists of a set of iterations which are similar to a project using one of the previously described waterfall models. The aim here is to have smaller subprojects which can be developed on their own in order to deliver a functional subontology which can be used in any other context. This approach is convenient if there is the need for a highly flexible environment to cope with changes in requirements or when the domain under consideration is very complex or if the project using the model is a rather large one. The benefits of this model are the chance to cope with possible risks in a stage as early as possible and to be flexible and open for changes in requirements. In each iteration new requirements can be added or existing ones can be altered/removed.

The set of requirements gets modified and improved iteratively based on the partial result of the previous iteration. As with the waterfall models the estimations of needed person months could be used

as basis of how long a subproject may take. With this estimations the project manager can set up a time line for each iteration of the subproject and furthermore for the whole ontology project. If there is a change in the requirement specification (one of the subprojects has to meet new requirements or some requirements will drop away) ONTOCOM will deliver new estimations which will affect the time line of the ontology project. This may reveal unneeded capacities or will reveal an issue with meeting release dates in an early stage so that more resources can be allocated.

gOntt

gOntt is a plug-in for the NeOn toolkit which provides additional functionalities focusing on project scheduling and supporting the ontology project execution.

Scheduling of ontology projects

When scheduling an ontology project the user has to make several decisions regarding the choice of a project development methodology (NeOn provides the waterfall and the iterative-incremental life cycle models), the order of activities, processes and phases of the chosen life cycle themselves, time restrictions and others. Each of these decisions can be supported by gOntt by providing the user with intuitive questions in order to make the appropriate decisions. Moreover gOntt presents the project development in a graphical manner using Gantt charts. When creating a new ontology project the user can either do this from scratch or in a guided way. Building a project from scratch implies that the user has experience in building ontology projects as well as knowledge of the important aspects of an ontology project. This is necessary since the user makes each decision on his own. This starts with the choice of which life cycle model to use. Then the user can add phases, processes and activities, order them according to a desired sequence, relate them among each other and he can add restrictions. In the guided way gOntt uses templates for scheduling ontology projects which serve as basis for the user's ontology project. By providing the user with questions regarding the project to schedule, the user can intuitively choose a life cycle model, as well as the needed phases, processes and activities. gOntt then produces a Gantt chart with the first draft of the project. Then it is up to the user to alter this plan by adding/altering/removing phases, processes and activities or by including new restrictions or relationships.

Ontology project execution support

The focus of this functionality is to provide the user with information for the execution of the project during its run-time. So called filling cards display important information to the user like the definition of a process or activity together with its inputs and outputs, its goal, time line and responsible actor and finally a work flow on how each of these aspects have to come together in order to fulfil the process or activity. Such a work flow is implemented with Eclipse Cheat Sheets. To each process or activity different plug-ins of the NeOn toolkit can be associated and triggered.

ONTOCOM/ONTOCOM II

ONTOCOM

ONTOCOM, which provides a tool to estimate the efforts involved in building, reusing and maintaining ontologies, is a generic cost model for ontology development, based on parametric-based cost estimation. The model is generic in that it assumes a sequential ontology life cycle, according to which, an ontology development project starts with the analysis of the domain and of the requirements to be fulfilled by the ontology; once an initial set of requirements is specified, the ontology is conceptualised, implemented and evaluated. The outcomes of the evaluation can lead to further iterations of the process either at the domain analysis, the conceptualisation or the implementation phase. The cost estimation model of ONTOCOM is realised as follows. First, a work breakdown structure for ontology development processes is defined in order to reduce the complexity of project budgetary planning and controlling operations down to more manageable units. The associated global effort estimates are then calculated using a parameterised mathematical method. The parameters of this formula are given start values in predefined 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 ONTOCOM cost drivers, which are proved to have a direct impact on the total development efforts, can be divided into three categories:

- **Product-related cost drivers** that account for the impact of the characteristics of the ontology on the overall costs. The following cost drivers were identified for the task of ontology development: Cost drivers for ontology building: Complexity of the Domain Analysis (DCPLX), Complexity of the Conceptualisation (CCPLX), Complexity of the Implementation (ICPLX), Complexity of the Instantiation (DATA), Required Reusability (REUSE), Documentation Needs (DOCU), Complexity of the Ontology Integration (OI), Complexity of the Ontology Evaluation (OE), Cost drivers for reuse and maintenance: Complexity of the Ontology Modifications (OM), Ontology Translation (OT), Ontology Understanding (OU), Ontologist/Domain Expert Unfamiliarity (UNFM)
- **Personnel-related cost drivers** that emphasize the role of team experience, ability and continuity with respect to the effort required by an ontology development project. In this category, the following cost drivers have been identified: Ontologist/Domain Expert Capability (OCAP/DECAP), Ontologist/Domain Expert Experience (OEXP/DEEXP), Personnel Continuity (PCON), Language and Tool Experience (LEXP / TEXP)
- **Project-related cost drivers** that are related to the characteristics of the overall ontology development project and their impact on the total costs. Relevant cost drivers include: Tool Support (TOOL), Multisite Development (SITE), Required Development Schedule (SCED)

The ONTOCOM cost drivers were defined after extensively surveying ontology engineering literature from the last decade and conducting expert interviews, and from empirical findings of numerous case studies in the field. For each cost driver we specified in detail the decision criteria which have to be

taken into account by the user of the model in order to determine the concrete rating level of the driver in a particular project setting.

ONTOCOM II

What one should be aware of with the cost estimation using ONTOCOM, as it has been the case with other software development cost estimation, is to realise that the estimation accuracy appears to heavily depend on a set of input values that describe the characteristics of the ontology itself and the related development environment. A significant part of the estimation failures can be attributed to a lack of understanding of these factors. In addition, one should recognise the purpose of the cost estimation. Generally, the aims of cost estimation are budgeting, and project planning and control as well as investment and risk analysis. Since the cost estimation will be performed frequently by means of various aims during the project, one can interpret the estimate result with the consideration of the aim and project environment. In other words, two simple guidelines that can improve the ontology cost estimation process should be used when applying ONTOCOM II.

Guideline 1: Keep all information related to the effort measurements.

The scope, level of conceptualisation and other assumptions of the effort estimate should be recorded. The version of ORSD and other documents that the effort estimate is based on should be specified. They should record what criteria are used for and how the values of cost drivers are derived. These records are of vital importance for accurate decision making regarding the input values and can be used as source documents whenever new estimations are required.

Guideline 2: When assessing estimation accuracy, make sure that the estimated and the actual effort are comparable.

If there is a noticeable difference between the estimated result and the actual efforts, the actual efforts should be re-examined so that they are comparable to the estimated result with respect to the core cost drivers. When estimates cannot be reliably matched to values that are comparable to the actual efforts, great care should be taken when using these results. In this case the project should be investigated and the re-estimation should be accomplished with the newly calibrated project scope and data.

The following estimation process refers to which activities are performed when estimating the cost required in ontology development. This process guides the proper application of ONTOCOM II:

1. Step 1 - Acquire and understand domain knowledge
2. Step 2 - Analyse functional requirements of the ontology development project
3. Step 3 - Estimate the effective size of the ontology
4. Step 4 - Estimate ontology development efforts
5. Step 5 - Estimate ontology reuse and maintenance efforts
6. Step 6 - Analyse available personnel supports
7. Step 7 - Estimate project and schedule efforts

Alignment of cost drivers to the NeOn glossary

In the course of this alignment we identified a series of issues related to the definition of cost drivers in ONTOCOM II:

- The need for a more detailed account of project management costs, covering NeOn activities such as feasibility study, scheduling and configuration management.
- The need for ontology location support as part of our ontology reuse cost drivers group. This will be implemented as a new cost driver as well.
- Discrepancies in the level of granularity of certain cost drivers, in particular ontology modification and ontology translation. These should be merged in a next version of the model.
- Methodologically sub-optimal support for knowledge reuse, which includes all aspects related to leveraging non-ontological resources in an ontology engineering project.

Product-related cost drivers

Cost drivers for ontology building

Complexity of the domain analysis (DCPLX)

The following list shows all the activities from the NEON glossary which refer to the term “domain complexity analysis” from the ONTOCOM model:

- Ontology Elicitation
- Ontology Specification
- Knowledge Acquisition For Ontologies

Complexity of the conceptualisation (CCPLX)

The following list shows all the activities from the NEON glossary which refer to the term “conceptualisation” from the ONTOCOM model:

- Ontology Conceptualisation

Complexity of the implementation (ICPLX)

The following list shows all the activities from the NEON glossary which refer to the term “implementation” from the ONTOCOM model:

- Ontology Formalisation
- Ontology Implementation

Complexity of the instantiation (DATA)

The following list shows all the activities from the NEON glossary which refer to the term “instantiation” from the ONTOCOM model:

- Ontology Population

Documentation needs (DOCU)

The following list shows all the activities from the NEON glossary which refer to “documentation needs” from the ONTOCOM model:

- Ontology Annotation
- Ontology Documentation
- Ontology Summarisation

Complexity of the ontology integration (OI)

The following list shows all the activities from the NEON glossary which refer to the term “ontology integration” from the ONTOCOM model:

- Ontology Aligning
- Ontology Comparison
- Ontology Integration
- Ontology Mapping
- Ontology Matching
- Ontology Merging

Complexity of the ontology evaluation (OE)

The following list shows all the activities from the NEON glossary which refer to the term “ontology evaluation” from the ONTOCOM model:

- Ontology Assessment
- Ontology Diagnosis
- Ontology Evaluation
- Ontology Repair
- Ontology Validation
- Ontology Verification

Cost drivers for reuse and maintenance

Complexity of the Ontology Modifications (OM)

The following list shows all the activities from the NEON glossary which refer to the term “ontology modification” from the ONTOCOM model:

- Ontology Customisation
- Ontology Enrichment
- Ontology Evolution
- Ontology Extension
- Ontology Forward Engineering
- Ontology Localisation
- Ontology Modularisation
- Ontology Module Extraction

- Ontology Partitioning
- Ontology Pruning
- Ontology Restructuring
- Ontology Specialisation
- Ontology Translation
- Ontology Update
- Ontology Upgrade
- Ontology Versioning

Personnel-related cost drivers

Personnel-related cost drivers emphasize the role of team experience, ability and continuity with respect to the effort required by an ontology development project. For this category there are no corresponding activities in the NeOn glossary. The activities of this glossary are about ontologies themselves, about steps that have to be performed for creating an ontology, steps that have to be performed to evolve an ontology and many other steps that have to be performed in order to develop an ontology one can benefit from. Due to this fact our adaption of the ONTOCOM model does not cover personnel-related cost drivers.

Project-related cost drivers

Project-related cost drivers are related to the characteristics of the overall ontology development project and their impact on the total costs. For this category there are no corresponding activities in the NeOn glossary. The activities of this glossary are about ontologies themselves, about steps that have to be performed for creating an ontology, steps that have to be performed to evolve an ontology and many other steps that have to be performed in order to develop an ontology one can benefit from. Due to this fact our adaption of the ONTOCOM model does not cover project-related cost drivers.

New cost drivers for ONTOCOM

The NEON glossary contains activities which cannot be mapped to an existing cost driver from the ONTOCOM model since these activities cannot be addressed by one of the existing cost drivers. Therefore we alter the ONTOCOM model by introducing new cost drivers (complexity of knowledge reuse and complexity of the project management) which then cover all those activities that could not be mapped to the ONTOCOM model. Furthermore we have to introduce new effort multipliers and corresponding rating scales:

The following list shows all the activities from the NEON glossary which refer to new cost driver “knowledge reuse” from the ONTOCOM model:

Complexity of knowledge reuse (KR)

- Non Ontological Resource Reengineering
- Non Ontological Resource Reuse
- Ontology Learning
- Ontology Reuse

- Ontology Reverse Engineering
- Ontology Search
- Ontology Selection

Rating	Rating Scale
High	the number of reused resources is very high as well as the effort in reusing them
Nominal	the number of reused resources is average and the effort for reusing them is moderate
Low	there are hardly resources to be reused and the corresponding effort is rather low

Project management (PM)

The following list shows all the activities from the NEON glossary which refer to new cost driver “project management” from the ONTOCOM model:

- Control
- Ontology Configuration Management
- Ontology Environment Study
- Ontology Feasibility Study
- Ontology Quality Assurance
- Scheduling

Rating	Rating Scale
High	the project management is highly intense, highly important and needs a lot of expertise
Nominal	the project management effort is average and the needed management skills are moderate
Low	the project management plays a minor role, only little expertise is needed

Applying ONTOCOM to the NeOn life cycle models

In the following we want to explain how ONTOCOM can support the ontology development project. Therefore we consider the different phases of the various types of the waterfall model with respect to ONTOCOM’s cost drivers and whether those can be used for planning and controlling the development.

As some of the phases are part of more than one waterfall model derivative we focus on the phases rather than on the specific model derivatives:

Initiation phase

The main purpose of the initiation phase (which is part of all the waterfall model derivatives) is to obtain a document which clearly covers the requirements we want the ontology to satisfy by considering knowledge related to the domain to model. After this initial activity the project has to be approved or rejected before setting up a development team, a time frame, a resource distribution and a

responsibility distribution. Essentially this first phase is about documentation as well as about project management aspects which is why we can use ONTOCOM's cost drivers DOCU and PM to plan the specific activities as well as to control their execution regarding adherence to requirements and compliance with schedules.

Design phase (Modelling phase)

The goal of the design phase, which is also part of more than one waterfall model derivative, is to obtain models (formal- and informal ones) which adhere to the requirements set up in the previous phase. Informal models serve as basis for better understanding and for further ontology development activities whereas formal models at this stage can be used in other ontology networks. The creation of both types of models can be supported by ONTOCOM in terms of planning and controlling since the cost drivers DCPLX and CCPLX can be used to estimate design related costs.

Implementation

Like the initiation phase, the implementation phase is part of each waterfall model derivative as it is about creating an ontology implemented in a language like RDF(S), OWL or another ontology representation language which can be processed by semantic applications or other ontology networks. As basis for the implementation we use the formal model we get from the design phase. Depending on the ontology development toolkit one uses the design phase and the implementation phase are carried out in parallel, yet we can use ONTOCOM to support the implementation phase. Therefore we use the cost driver ICPLX which provides estimations about the effort arising from performing the necessary activities. We can also use this cost driver to control the implementation phase when comparing the estimations to the real progress of this phase.

Maintenance phase

Like the initiation phase and the implementation phase, the maintenance phase is part of each waterfall model derivative. It is about detecting errors as well as missing knowledge, which can be detected during the use of an ontology, and subsequently it is also about coping with the mentioned issues (as this phase can restart the design phase). An important aspect of this phase is to generate new versions of the ontology under consideration which can be planned and controlled with the cost driver OM.

Reuse phase

During an ontology development project we need not always to start from scratch. We should rather consider the reuse of already implemented ontological resources. Therefore we search for and select among available ontology resources to obtain models (formal- and informal ones) which then can be used in the modelling phase. Furthermore we could also use an implemented model (adhering to some kind of ontology representation language) in the implementation phase. The corresponding activities can be planned using the cost driver KR and of course we can use the resulting estimations as basis for controlling the activities as we can get an idea of whether the execution is on time or not.

Merging phase

The merging phase is only part of the extended waterfall model derivatives (5+, 6+). Its purpose depends on the specific model. In the extended five-phase model we initiate this phase after we have completed the reuse phase in order to obtain a new ontological resource. This can be done by merging two or more

ontological resources which we found in the reuse phase. In the extended six-phase model we initiate this phase after we have completed the reengineering phase (see below) in order to merge ontological as well as non ontological resources which we found in the reengineering phase. In both models we can use the cost driver OI to estimate the effort necessary for performing the specific activities which supports the development of an ontology regarding planning and controlling.

Reengineering phase

The reengineering phase is about reengineering non ontological resources (where we transform a non ontological resource in an ontological resource) which can be detected by initiating the reuse phase. In addition the development team has to decide whether or not to reengineer also ontological resources. The overall goal is to make existing resources (ontological ones and non ontological ones) available for the ongoing ontology development project. This reengineering phase can be planned and its execution can be controlled by using ONTOCOM's cost driver OE.

How to benefit from ONTOCOM estimations

In order to enrich gOntt with cost estimation capabilities it is necessary to provide tool support in terms of providing access to the ONTOCOM II model during scheduling of ontology projects. We planned to alter the gOntt scheduling interfaces for applying the guided way as well as for the ontology creation from scratch in a way that the ontology project manager can access the ONTOCOM II web interface for estimating costs. In both cases the ontology project manager has to choose an entry from the provided rating scales for the given cost drivers which results in the overall costs for the ontology development project. Based on his expertise and previously completed ontology projects the project manager then as to distribute the mentioned overall costs to phases. This can be done for example by distributing costs to the phases as percentages of the overall costs. After associating personnel to the phases and activities the result of this procedure is a detailed time line describing how much time is needed to complete a phases as well as for completing the whole project.

Let us assume that according to the ontology to develop we have the following inputs for ONTOCOM II and the respective output as result from the calculation:

Ontology Size

Product	The product category accounts for the influence of product properties on the overall costs			
	dcplx <input type="text" value="Very High"/>	ccplx <input type="text" value="Very High"/>	icplx <input type="text" value="Very High"/>	pm <input type="text" value="Normal"/>
	reuse <input type="text" value="Normal"/>	docu <input type="text" value="Very High"/>		
Reuse	The reuse category accounts for the influence of reuse costs			
	oe <input type="text" value="Normal"/>	om <input type="text" value="Normal"/>	kr <input type="text" value="Normal"/>	oi <input type="text" value="Normal"/>

Estimated Benefit: 10.06835661173294

This would mean that the overall costs of the ontology project will be around 10 person months which have to be distributed to the different phases of the NeOn waterfall models. In case that we use the simplest version of the waterfall derivatives we would have to assign these 10 person months to the four phases *Initiation Phase*, *Design Phase*, *Implementation Phase* and *Maintenance Phase*. Let us assume that the past ontology projects have shown that most of the work has to be done in the Implementation Phase, followed by the Maintenance Phase, followed by the Design Phase and the Initiation Phase. Then we could distribute the overall effort to the phases as follows:

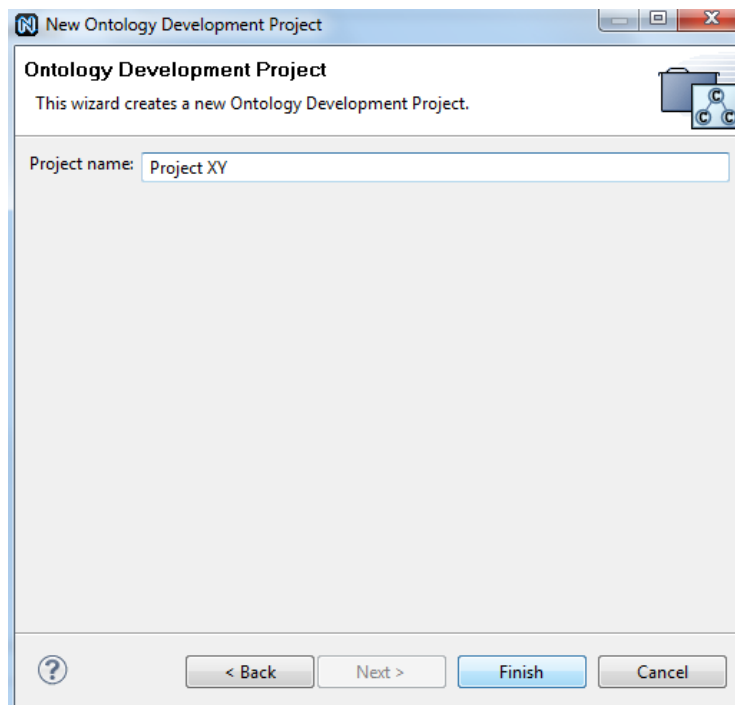
	Percentage of overall effort (10PM)	Person months/phase
Initiation Phase	10%	1
Design Phase	20%	2
Implementation Phase	40%	4
Maintenance Phase	30%	3

If we use another waterfall derivative then we simply have more phases we have to assign person months to. Again this is based on expertise of the project manager.

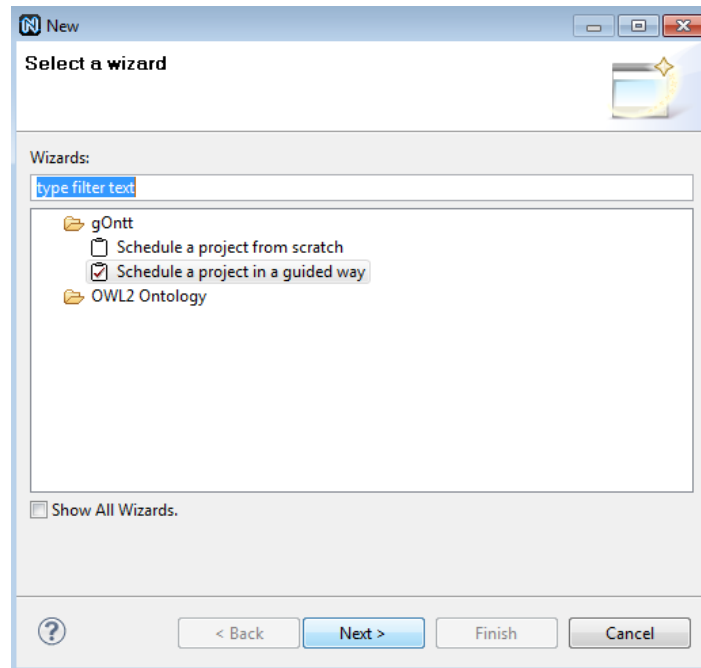
Implementation mock-up

The planned installation will be about providing access to ONTOCOM II during the scheduling of ontology projects. Therefore we use for example the interface of the guided tour and add some more decisions to take, input fields and text fields. The process of scheduling a project in the guided tour will then look as following:

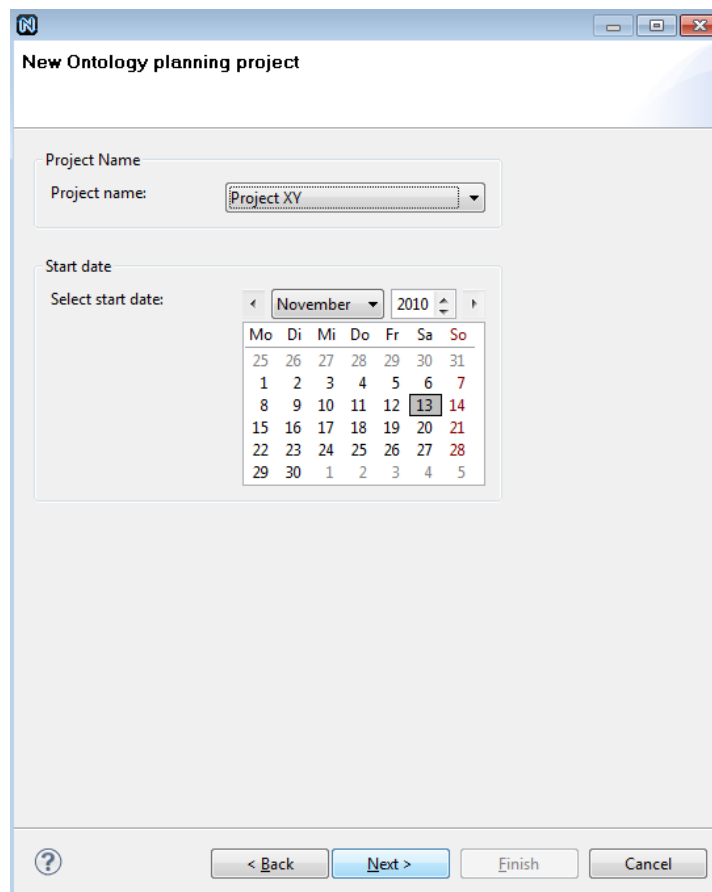
First of all we create an ontology project within the NeOn development toolkit:



Then we create a gOntt schedule e.g. in the guided way:



We assign an ontology development project and a start date:



Since we know all requirements of the ontology to create we can choose the corresponding radio button:

New Ontology planning project
The ontology life cycle model.

Are the ontology requirements assumed to be fully known at the beginning of the ontology network development?

Yes No

? < Back Next > Finish Cancel

Then we answer the questions which are provided by gOntt in order to support the development:

New Ontology planning project
Waterfall life cycle

Scenario 1: From specification to implementation. Yes No

Scenario 2: Have you planned to use any non-ontological resource such as thesauri, data bases, etc. in your ontology network development? Yes No

Scenario 3: Have you planned to use any existing ontological resource in your ontology network development? Yes No

Scenario 4: Have you planned to use and modify any existing ontological resource in your ontology network development? Yes No

Scenario 5: Have you planned to use and merge a set of existing ontological resources in your ontology network development? Yes No

Scenario 6: Have you planned to use, merge, and modify a set of existing ontological resources in your ontology network development? Yes No

Scenario 7: Have you planned to use ontology design patterns in your ontology network development? Yes No

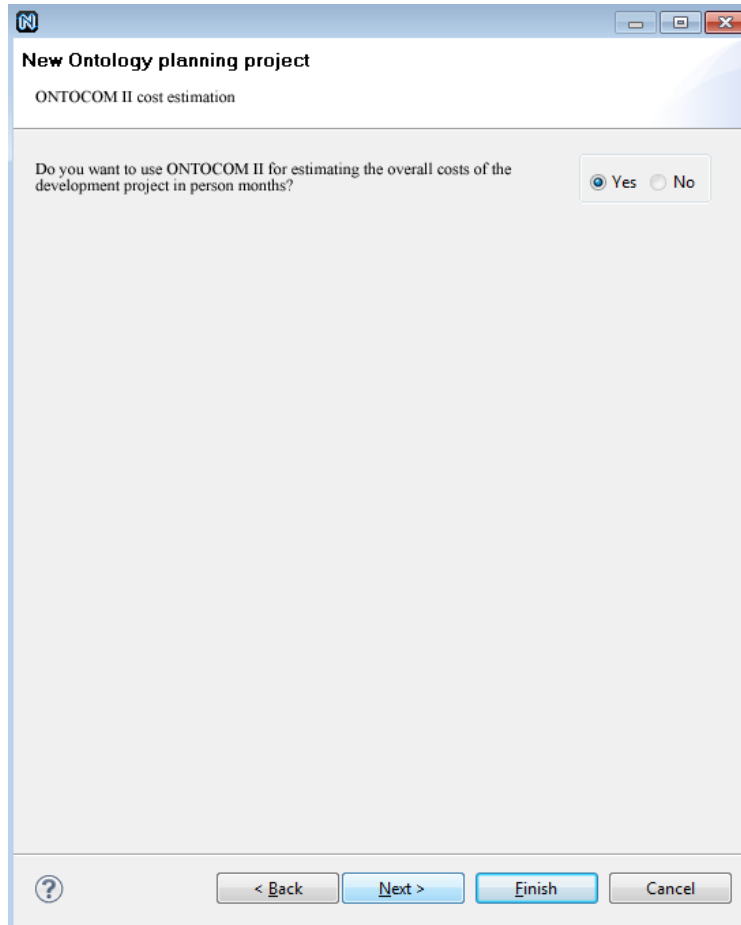
Scenario 8: Have you planned to restructure your ontology network? Yes No

Scenario 9: Have you planned to develop your ontology network in different natural languages? Yes No

[See Picture](#)

? < Back Next > Finish Cancel

After answering all the questions we get to a new part of the interface where we can choose to use ONTOCOM II or not:



In case that we do not do so the project schedule will be created as usual. If we choose to use ONTOCOM II we will be directed to the ONTOCOM II web interface, where we have to choose values regarding the impact of the provided cost drivers which will result in the overall effort in person months:

Ontology Size

The product category accounts for the influence of product properties on the overall costs					
Product	dcplx	Very High	ccplx	Very High	
	icplx	Very High	pm	Normal	
	reuse	Normal	docu	Very High	
The reuse category accounts for the influence of reuse costs					
Reuse	oe	Normal	om	Normal	
	kr	Normal	oi	Normal	

Estimated Benefit: 10.06835661173294

In another new part of the interface the project manager has to fill in the overall effort for the project. For the different phases he additionally has to distribute the overall effort in terms of percentages:

The screenshot shows a software window titled "New Ontology planning project". The window contains the following elements:

- A header bar with a logo on the left and standard window control buttons (minimize, maximize, close) on the right.
- A main content area with the following text and input fields:
 - "Please fill in the result of the ONTOCOM II calculation:" followed by a text box containing the value "10.06835661173294".
 - "Please fill in the distribution of the overall effort in percentage to the different phases:"
 - "Initiation Phase:" followed by a text box containing "10%".
 - "Design Phase:" followed by a text box containing "20%".
 - "Implementation Phase:" followed by a text box containing "40%".
 - "Maintenance Phase:" followed by a text box containing "30%".
- A footer bar containing a help icon (question mark in a circle) on the left, and four buttons: "< Back", "Next >", "Finish", and "Cancel".

After clicking on "Finish" the project manager gets provided with the initial development schedule including estimations for the different phases:

