gOntt: a Tool for Scheduling Ontology Development Projects

Asunción Gómez-Perez
Ontology Engineering Group
(Universidad Politécnica de Madrid)
Campus de Montegancedo sn.
28660. Boadilla del Monte (Spain)
+34 91 336 7439
asun@fi.upm.es

Mari Carmen Suárez-Figueroa
Ontology Engineering Group
(Universidad Politécnica de Madrid)
Campus de Montegancedo sn.
28660. Boadilla del Monte (Spain)
+34 91 336 3672
mcsuarez@fi.upm.es

Martín Vigo
Ontology Engineering Group
(Universidad Politécnica de Madrid)
Campus de Montegancedo sn.
28660. Boadilla del Monte (Spain)
+34 91 336 3670
mvigo@delicias.dia.fi.upm.es

ABSTRACT

The Ontology Engineering field lacks tools that guide ontology developers to plan and schedule their ontology development projects. gOntt helps ontology developers in two ways: (a) to schedule ontology projects; and (b) to execute such projects based on the schedule and using the NeOn Methodology [2].

Categories and Subject Descriptors

D.2.3 [Coding Tools and Techniques]: *Object-oriented programming*; D.2.6 [Programming Environments]: *Integrated environments*; D.2.9 [Management]: *Life Cycle*.

General Terms

Algorithms, Management, and Design.

Keywords

Ontology management, ontology life cycle, scheduling.

1. INTRODUCTION

Planning and scheduling are related activities that are applied in different settings such as civil engineering, software development, etc. While planning[1] is the act of drawing up plans, which are a series of steps to be carried out to achieve an objective, scheduling[1] is defined as the activity to set order and time to planned events. Scheduling should be performed after planning; and both are crucial in any project.

In the Software Engineering field, every development project is planed and scheduled before starting it. The project plan defines the tasks to be done and the human resources to perform the project work. The project schedule is a calendar that links the tasks to be done with the resources to support their performance. The most common form of schedules is a Gantt chart [1]; and the most popular tool for creating a project schedule is Microsoft Office Project [1]. Microsoft Project provides three different ways to create project schedules: (a) from scratch by adding tasks, phases, etc.; (b) from a project template (engineering template, commercial construction template, annual report preparation template, etc.) selected by the user from a template library; and (c) from an existing project. However, from our knowledge, any existing planning tool from creating and managing project schedules provides developers with guidelines on how to execute the project.

Unlike what happens in Software Engineering, in the Ontology Engineering field planning and scheduling ontology developments are still in their early stages. This field lacks methods and tools to guide ontology developers in planning and scheduling their ontology projects. Additionally, (a) there is no project templates oriented to schedule ontology developments in the Microsoft Project template library, (b) there is no support tool for providing ontology developers with ontology project schedules in the form of Gantt charts, and (c) there is no support tool that recommend the ontology developer which methodological guidelines and tools should be used for executing a process or an activity in the ontology development.

2. gOntt MAIN FEATURES

gOntt (see Figure 1) overcomes the aforementioned limitations by (a) using project templates oriented to schedule ontology developments based on the scenarios proposed by the NeOn Methodology [2], (b) generating the scheduling of ontology developments in the form of Gantt charts, and (c) informing ontology developers about how to carry out a process or an activity using the NeOn Methodology prescriptive guidelines and about the concrete NeOn Toolkit[2] plug-in to be used. Next, we describe gOntt functionalities for scheduling ontology projects and for helping in the execution of ontology projects.

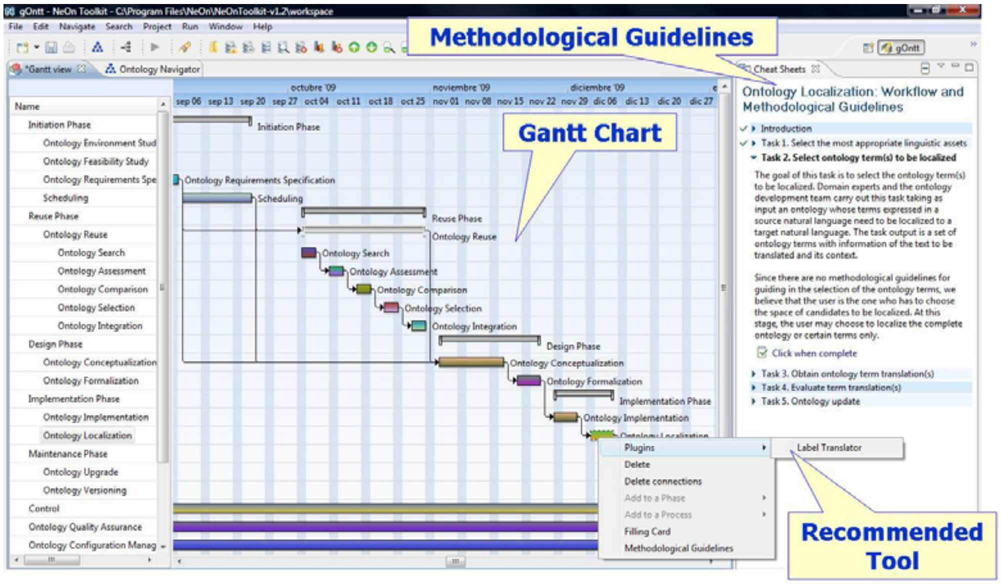
2.1 Scheduling Ontology Projects

gOntt provides support to ontology developers (a) to decide which ontology life cycle model is the most appropriate for building their ontologies (e.g., waterfall, iterative-incremental), which processes and activities should be carried out and in which order (e.g., specify ontology requirements before re-engineering a knowledge-aware resource into an ontology) and (b) to create a graphical representation in the form of a Gantt chart with the processes and activities needed, including time restrictions between them. Schedules for ontology development projects can be created either *from scratch* or *in a guided way*.

In the first case, gOntt allows the developer to include processes, activities, phases, and relationships and restrictions among them according to his/her needs.

[1] http://wordnetweb.princeton.edu/perl/webwn

[2] http://www.neon-toolkit.org/



Ontology Localization: Workflow and Methodological Guidelines

✓ Introduction

✓ Task 1. Select the most appropriate linguistic assets

Task 2. Select ontology term(s) to be localized

The goal of this task is to select the ontology term(s) to be localized. Domain experts and the ontology development team carry out this task taking as input an ontology whose terms expressed in a source natural language need to be localized to a target natural language. The task output is a set of ontology terms with information of the text to be translated and its context.

Since there are no methodological guidelines for guiding in the selection of the ontology terms, we believe that the user is the one who has to choose the space of candidates to be localized. At this stage, the user may choose to localize the complete ontology or certain terms only.

☑ Click when complete

Task 3. Obtain ontology term translation(s)

Task 4. Evaluate term translation(s)

Task 5. Ontology update

Figure 1. Screenshot of gOntt

In the guided way, gOntt creates a preliminary plan for the ontology development using templates for scheduling ontology projects and a simple two-step wizard that contains simple and intuitive questions that implicitly allow the ontology developer to select the ontology life cycle model and the processes and activities needed in his/her development. gOntt internally uses two decision trees and a set of templates to automatically generate the initial plan. Templates were built taking into account the NeOn Methodology for building collaboratively large scale ontologies by means of reuse and re-engineering knowledge- resources. gOntt main output is the initial plan for building the ontology in the form of a Gantt chart that the developer can modify later on (a) by including, modifying, or deleting processes, activities, and phases, (b) by changing order and dependencies among processes and activities, and (c) by including resource assignments and restrictions to the plan.

2.2 Helping in Ontology Project Executions

With the aim of helping the ontology developer to carry out a particular process or activity, gOntt provides ***prescriptive methodological guidelines*** by means of: (1) a **filling card** that includes the process or activity definition, its goal, inputs and outputs, who carries it out, and when it should be done, and (2) a **workflow** describing how the process or the activity should be done with its inputs, outputs, tasks and actors involved. Workflows are implemented with Eclipse Cheat Sheets[3].

At this moment, gOntt supports the following processes and activities of the NeOn Methodology: ontology requirements specification, scheduling, reusing and re-engineering non-ontological resources, reusing ontological resources, reusing ontology design patterns, ontology localization, ontology modularization, ontology evaluation and ontology evolution.

In addition, gOntt provides a direct access to NeOn Toolkit plug-ins associated to each process and activity as can be seen in Figure 1. This means that gOntt triggers the different NeOn Toolkit plug-ins associated to each process or activity included in the plan.

3. CONCLUSIONS

gOntt is the first tool created so far for systematizing the planning and scheduling of ontology projects. It uses templates and decision trees to create initial ontology project schedules in the form of a Gantt chart. In addition, gOntt is the first meta-tool that helps ontology developers to execute ontology projects. It includes prescriptive methodological guidelines and recommended tool to be used for executing a particular process or activity in an ontology development project. gOntt has been developed in Java[4] as a plug-in for the NeOn Toolkit.

In a nutshell, the most important difference between gOntt and Microsoft Project is that while in gOntt the system generates automatically the initial ontology development plan in Microsoft Project the user selects a project template from the library. Furthermore, there is no planning tool from creating and managing project schedules that provides developers with guidelines on how to execute the project as gOntt does.

4. ACKNOWLEDGMENTS

This work has been supported by the project NeOn (FP6-027595). We are very grateful to Elena Montiel-Ponsoda.

5. REFERENCES

[1] Stellman, A. and Greene, J. 2006. Applied Software Project Management. ISBN: 0-596-00948-8.

[2] Gómez-Pérez, A. and Suárez-Figueroa, M.C. 2008. NeOn Methodology: Scenarios for Building Networks of Ontologies. 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW 2008). Conference Poster. Italy.

[3] http://www.ibm.com/developerworks/opensource/library/os-ecl-cheatsheets

[4] Platform J2SE, version 5.0