apodle learn @ work

Information Society Technologies

Project Number **027023**

APOSDLE: Advanced Process Oriented Self-Directed Learning Environment

Integrated Project

IST – Technology enhanced Learning

Integrated Modelling Methodology Version 1

Deliverable D 1.3

Due date	**2007-11-30**
Actual submission date	**2007-11-28**
Start date of project	2006-03-01
Duration	48
Revision	**Final**

Organisation name of lead contractor for this deliverable

FBK – FONDAZIONE BRUNO KESSLER

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level

☒	PU	Public
☐	PP	Restricted to other programme participants (including the Commission Services)
☐	RE	Restricted to a group specified by the consortium (including the Commission Services)
☐	CO	Confidential, only for members of the consortium (including the Commission Services)

Disclaimer

Imprint

Full project title:	Advanced Process-Oriented Self-Directed Learning Environment
Title of work package:	WP 1: Work Processes
Document title:	Integrated Modelling Methodology Version 1
Document Identifier:	APOSDLE-D01.3-FBK-IntegratedModellingMethodology1
Work package leader:	SAP
List of authors:	Chiara Ghidini, Marco Rospocher, Luciano Serafini (FBK)
	Barbara Kump, Viktoria Pammer (TUG)
	Andreas Faatz (SAP)
	Joanna Guss (EADS IW)
Administrative Co-ordinator:	Harald Mayer
Scientific Co-ordinator:	Stefanie Lindstaedt

Copyright notice

Document History

Version	Date	Reason of change
1	2007-10-18	Document created
2	2007-11-12	All contributions inserted. Revised by Ghidini
3	2007-11-15	Executive summary added. Appendix with Feedback questionnaires added
4	2007-11-15	Evaluation section added
5	2007-11-19	Final revision before submission to internal reviewers
6	2007-11-27	Final revision after internal review
7	2007-11-28	Submitted version

Executive Summary

This document describes the APOSDLE Integrated Modelling Methodology. This methodology, developed within the APOSDLE project, guides the process of creation of the application domain dependent parts of the APOSDLE Knowledge Base. The APOSDLE Knowledge Base provides the basis for reasoning within the APOSDLE System.

The methodology consists of five distinct phases, which cover the entire process of model creation, from the initial selection of the application domain, to its informal, and then formal, specification, and finally to the validation and revision of the final Knowledge Base:

- **Phase 0. Scope & Boundaries**. In this phase the scope and boundaries of the application domain are determined and documented. The first step of this phase was to use questionnaires to elicit the main tasks and learning needs of the different Application Partners in order to identify candidate application domains for learning (also called learning domains). The candidate application domains were then discussed and the final domain was decided upon, and briefly documented. The key aspect of this phase is to support the Application Partners to identify a learning domain appropriate for the "learn @ work" approach taken by APOSDLE.
- **Phase 1. Knowledge Acquisition**. The goal of this phase is the acquisition of knowledge about the application domains that have to be formalised and integrated in the APOSDLE knowledge base. The proposed methodology aims to extract as much knowledge as possible from both Domain Experts and available digital resources identified by the Application Partners. The elicitation of knowledge from Domain Experts is based on well known techniques like interviews, card sorting and laddering, already introduced in literature, while the extraction of knowledge from digital resources is based on algorithms and tools for term extraction described in (Pammer, Schier, & Lindstaedt, 2007). The key aspect of this phase is twofold: first, the methodology has to support an effective and rapid knowledge acquisition from Domain Experts, who are often rarely available and scarcely motivated towards modelling; second the methodology has to ease the process of modelling by reusing knowledge already present in digital format in the organisation.
- **Phase 2. Informal Modelling**. The goal of this phase is to start from the knowledge elicited in Phase 1 and provide an informal but structured and rather complete description of the different models which will constitute the APOSDLE knowledge base. These models concern (i) the processes and tasks a user can perform in the organisation, (ii) the specific domain of affaires (application domain) a user want to learn about with APOSDLE, and (iii) the learning goals a user can have in the organisation inside the specific domain of affaires. The descriptions of the informal models are obtaining by filling pre-defined templates provided in a Semantic Wiki. The use of a Semantic Wiki allows to describe the elements of the different models in an informal manner using the Natural Language. However, at the same time it allows to structure the descriptions so that they can be easily (and often automatically) translated in formal models, without forcing the Application Partners to become experts in the formal languages used to produce the formal models. The key aspect of the informal modelling is to provide the Application Partners with tools that support informal modelling – and thus hide the complexity of formal languages – but at the same time are not mere textual editors and provide some typical facilities of modelling tools, such as the possibility to have a (possibly graphical) overview of the entire model.
- **Phase 3. Formal Modelling and Integration**. In this phase the informal descriptions of the task model and of the domain model are transformed in formal models. The current version of the methodology supports the automatic translation of the domain model in an OWL ontology, and a manual translation of the task model in a YAWL workflow. Additionally, the information about the learning goals contained in the informal models is automatically extracted and used

by the Task And Competency Tool (TACT for short), to help the Application Partners to formally specify learning goals. The output of this phase is a first version of the APOSDLE Knowledge Base. The key aspect of this phase is to ease the transformation from informal to formal as much as possible to avoid duplication of work and to make full use of the rich informal models produced during phase 2.

- **Phase 4. Validation & Revision**. In this phase the APOSDLE Knowledge Base is evaluated and possibly revised. The current version of the methodology provides the support for checking automatically, via SPARQL queries, different aspects (properties) of the APOSDLE knowledge base and of its single components. The results of these check are evaluated and used to revise the knowledge base, if needed. The key aspects of this phase are first to provide an accurate set of tests for the APSODLE knowledge base, and second to allow the Domain Experts to validate the models obtained. This second aspect is particularly challenging as the Domain Experts usually are not familiar with formal models.

This methodology has been accurately followed by the Application Partners to build their specific APOSDLE Knowledge Base. The specific models created are described in Deliverable D6.8 – Application Partner Domain Models.

Main results, coming from the definition and application of the Integrated Modelling Methodology for the production of the Application Partner Domain Models to be used in Prototype 2 are both positive and negative findings. On the positive side the development of the Integrated Modelling Methodology has allowed to instantiate the general task of “modelling” inside the APSODLE context, and to identify the key aspects and challenges of the modelling phase inside APOSDLE. In addition the application of the methodology has enabled the Application Partners to produce, in most of the cases, formal models of good quality, ready to be used in Prototype 2. On the negative side the development and application of the Integrated Modelling Methodology was not without problems. For this reason, and to help with the definition of the second version of the methodology, to be developed for Prototype 3, we have carried out a careful evaluation of the methodology with the help of the Application Partners. The results of this detailed evaluation, together with some suggestions for improvement, are included in the final part of this Deliverable.

Table of Contents

1 Introduction

1.1 Purpose of this document

This document describes the APOSDLE Integrated Modelling Methodology. This methodology, developed within the APOSDLE project, guides the process of creation of the application domain dependent parts of the APOSDLE Knowledge Base. The APOSDLE Knowledge Base provides the basis for reasoning within the APOSDLE System.

This methodology has been accurately followed by each Application Partner to build their specific APOSDLE Knowledge Base. The specific models created are described in Deliverable D6.8 – Application Partner Domain Models.

This document provides a detailed overview of the reasons behind the development of the Integrated Modelling Methodology (Section 2), an overview of existing methodologies for Ontology development (Section 3), a detailed description of the current version of the Integrated Modelling Methodology used by each Application Partner to build their specific APOSDLE Knowledge Base (Section 4), and finally an evaluation of the Methodology and some ideas for further enhancements (Section 5).

1.2 Related Documents

This deliverable is strictly related to the following documents:

- APOSDLE Deliverable D2.3 – Conceptual Framework & Architecture;
- APOSDLE Deliverable D6.8 – Application Partner Domain Models;
- APOSDLE Deliverable D4.3 – Software Architecture for 2nd APOSDLE Prototype;
- APOSDLE Semantic MediaWiki[1]: https://aposdle.fbk.eu/modelling/

[1] To access the wiki content, use the following login/password details: Username = reviewer, password = aposdle

2 Why a Modelling Methodology?

One of the key tasks when configuring and deploying the APSODLE System in a new domain/organization is the construction of the APSODLE Knowledge Base. The APOSDLE Knowledge Base is necessary for the APOSDLE System in order to allow the APOSDLE Platform to deliver to the APOSDLE User context-sensitive learning events, tailored to his/her specific learning goals, work situation and learning needs.

The APOSDLE Knowledge Base is composed of several models and the relations between them, as we can see with the help of Figure 1. The four main entities depicted in his figure are: a task, a learning goal, a domain model element and some specific APOSDLE categories. These entities are also related to each other with the specific relations depicted in the figure.

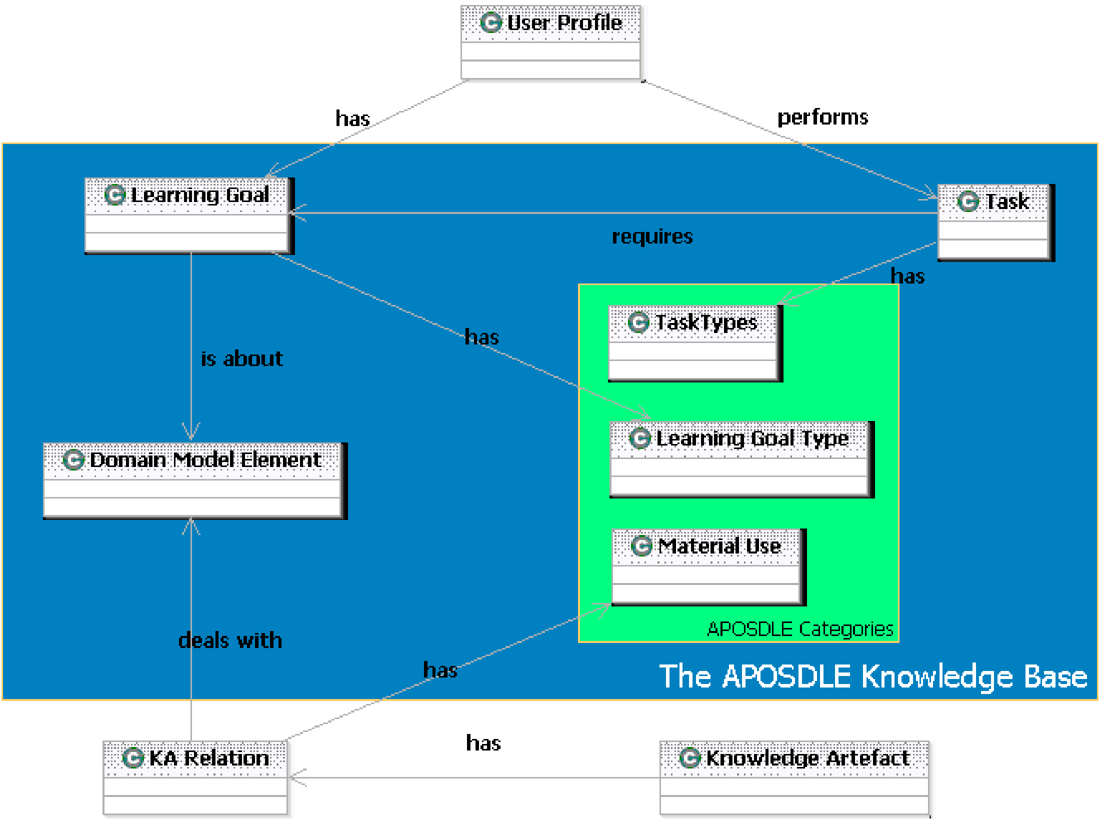

Figure 1. The APOSDLE Knowledge Base and its relations with the other APOSDLE components.

In a nutshell, the APOSDLE Knowledge Base contains all the necessary information about the tasks a user can perform in a certain organization, the learning goals required to perform certain tasks, a description of the domain of affairs (application domain) of the organization, and, finally, specific APOSDLE categories used to classify tasks, learning goals and learning material. The information about the tasks is contained in a task model, the information about the application domain is contained in the domain model, the information about the learning goals is contained in the learning goal model, and finally the categories used to classify tasks, learning goals and learning material are contained tin the APOSDLE categories. The main idea is that starting from the context-sensitive situation of a user, which includes her current task(s), ASPODLE is able to determine the learning goals of the user and then use the information contained in the APOSDLE Knowledge Base to select appropriate knowledge artifacts and transform them into learning events proposed to the user for attaining the missing learning goals. More details about the role of the APOSDLE Knowledge Base in the APOSDLE System can be found in Deliverable D2.3 (Conceptual Architecture).

Of the four models depicted in the blue rectangle of Figure 1 some are APOSDLE-inherent structures, while others are entirely application domain dependent. In particular the APOSDLE Categories are APOSDLE-inherent structures used by the system to classify, from a learning perspective, tasks, learning goals, and resources to be used as learning material, and are developed by the APOSDLE consortium. On the contrary, the Domain Model, the Task Model and the Learning Goal Model are by nature domain/organization dependent and need to be created every time the APOSDLE system is configured and deployed for a new organization / domain.

If we consider the typical application environment of APOSDLE, we can safely assume that most of the organizations interested in using the system won't have formal models of:

(i) their specific application domains,

(ii) their tasks and processes, and

(iii) the learning goals a user has to have / acquire

already available to be "plugged" in the APOSDLE system. On the contrary, it is likely the case that they will not even be interested in developing such models, as their main interest towards APOSDLE lies in setting up a tool that enhances the productivity of their work environment and offers to transform it into an integrated work-, learn- and collaboration environment. The situation is made even more complex by the fact that the models, which compose the APOSDLE Knowledge Base, are implemented using different data structures, which are the most appropriate to formalize the kind of knowledge they contain:

- tasks and processes are modelled using the workflow language YAWL;
- domain and learning goals are modelled using the ontology language OWL.

This would force the organization to become acquainted with more than one formal representation language.

Since the models are necessary for the functioning of the system, this let the APOSDLE consortium face an important challenge: to provide a methodology and tools to support and simplify the process of creation of the domain dependent models of the APOSDLE Knowledge Base.

To be effective and tailored to the APOSDLE system, such a methodology must satisfy some APOSDLE-inherent requirements:

1. provide the organization with high level tools useful to specify knowledge with natural language or in a graphical manner, to minimize the need to become familiar with the different formal languages;
2. support an integrated development of the three models (domain, tasks, and learning goals) in order ensure a conceptual consistency among these models and an easy formal integration;
3. support, in an homogeneous manner, the creation of formal models which are described using different languages (YAWL for the workflow based process representation of the task model, and OWL for the ontological representation of the domain model and learning goal models);
4. encourage knowledge engineers and domain experts from the organization to be the main actors of the modelling phase. In the event of unavailable knowledge engineers, or knowledge engineers not enough skilled to follow the process in an autonomous manner, provide some additional support and coaching from the APSODLE experts;
5. support a collaborative development of the APOSDLE knowledge base in order to allow the different actors (knowledge engineers, domain experts, coaches and APOSDLE experts) to work together on the development of the different models.

Although the literature abounds with methodologies, which cover different aspects of modelling, few of them are domain-independent and ready to be applied to the specific needs of APOSDLE. Furthermore, none of them fulfilled in a satisfactory way all the requirements we had. For this reason we have built upon existing methodologies for model development and have proposed a new methodology specific for the development of the domain dependent models of the APOSDLE Knowledge Base. In our review of the state of the art we have referred mainly to methodologies for ontology development, because we found them the most appropriate ones to support the modelling of the entire APOSDLE Knowledge Base in a uniform manner. Specific methodologies developed for the design of workflow processes have been considered for the modelling of tasks, and especially the formal modelling of tasks. In Section 3 we provide a brief overview of the state of the art methodologies for ontology development and workflow processes we have considered, while in Section 4 we provide a description of our methodology, and its usage to develop the APOSDLE Knowledge Bases for the 2nd Prototype of the APOSDLE system.

3 State of the Art

In this section one can find a brief overview of the most representative methodologies for ontology development. For a more in-depth survey of the state of the art, see (Cuel & Cristani, 2004), (Fernández-López, 1999), (Beck & Pinto, 2002) and (Roussey, 2005).

As observed in (Fernández-López, 1999), since ontologies are usually part of software products, a methodology for ontology development should follow the guidelines proposed in standards for software development, in particular the IEEE Standard for Developing Software Life Cycle Processes, 1074-1995 (IEEE Computer Society, 1996). However, none of the methodologies proposed up to now fully follows the guidelines of the IEEE standard (Fernández-López, 1999). On the contrary the work reported in (Roussey, 2005) allows to conclude that that, despite the differences, the modelling process proposed by all available methodologies is iterative and is based on the following basic steps:

1. **ontology specification**: the purpose and the domain of the ontology is determined;
2. **knowledge acquisition**: knowledge is (i) elicited from domain users/experts, and (ii) extracted from available resources describing the domain;
3. **conceptualization**: concepts and relations are detected, defined and organized (for example, look for relations between concepts);
4. **formalization**: build, using an appropriate language, a formal representation of the ontology;
5. **evaluation**: the ontology is evaluated with respect to the (i) satisfaction of the users when checking the ontology, (ii) completeness of the representation of the domain and (iii) correctness of the knowledge base;
6. **documentation**: choices performed and problems arisen during the process should be reported.

These steps are usually performed either sequentially, in parallel or in a combined way (some steps are performed in parallel, some steps are performed sequentially).

Uschold and King (Uschold & King, 1995) propose an ontology building process which is purely manual. Some guidelines are proposed to fulfil the basic steps of the methodology, which are:

1. identify the purpose;
2. building the ontology (which includes capturing, coding and integrating ontologies);
3. evaluation;
4. documentation.

This methodology has been applied for building the Enterprise Ontology[2], which is a collection of terms and definitions relevant to business enterprises.

Toronto Virtual Enterprises (TOVE) is a methodology proposed by Gruninger and Fox (Gruninger & Fox, 1994) as a result of their experience in the TOVE project. They introduced *competency questions* (a list of question that a knowledge base based on the ontology should be able to answer) as a technique to define the scope of the ontology. Their methodology also includes some techniques to deal with maintenance issues. Apart from the TOVE project, this methodology has been applied to build ontologies for different applications, including Enterprise Design Workbench and Integrated Supply Chain Management Project agents.

One of the most famous and used ontology building methodologies is METHONTOLOGY (Fernández, Gómez-Pérez, & Juristo, 1997). This methodology is inspired by the activities identified in the IEEE

[2] http://www.aiai.ed.ac.uk/project/enterprise/

Standard for Developing Software Life Cycle Processes. It enables the construction of ontologies at the *knowledge level* (that is, the conceptual level). It identifies three categories of activities:

- **Project management activities**, which include *planning* (identifies the tasks to be performed, how they will be arranged, etc.), *control* (guarantees completion of the planned tasks) and *quality assurance* (assures that the quality of the output is satisfactory) ;
- **Development-Oriented activities**, which include *specification* (why the ontology is being built, what are its intended uses and who are the users), *conceptualization* (modelling of the domain knowledge at the knowledge level), *formalization* (transformation of the conceptual model into a formal model) and *implementation* (building the models in a computational language).
- **Support activities**, performed at the same time as Development-Oriented activities, which include *knowledge acquisition* (acquisition of domain knowledge), *evaluation* (technical judgment of the ontologies, software products and documentation with respect to a frame of reference), *integration* (reusing other available ontologies) and *documentation* (reports on the steps completed and the output obtained).

This methodology has been applied to build ontologies such as CHEMICALS (Fernández M. , Gómez-Pérez, Sierra, & Sierra, 1999) (about chemical elements and crystalline structures), Environmental pollutants ontologies (Gómez-Pérez & Rojas-Amaya, 1999) (about methods of detecting pollutants), The Reference-Ontology (Aguado, et al., 1998) (an ontology in the domain of ontologies) and the $(KA)^2$ ontology (Blázquez, Fernández-López, García-Pinar, & Gómez-Pérez, 1998) (about the scientific community in the field of Knowledge Acquisition).

The OTK methodology (Sure & Studer, 2002) has been developed within the On-To-Knowledge project. It is a methodology that includes guidelines for introducing ontology-based knowledge management concepts and tools into enterprises. It is based on the CommonKADS methodology (Schreiber, et al., 2000) (a methodology to support structured knowledge engineering) and it is an application-driven ontology development process. It is based on the following steps:

- Feasibility study: serves as a decision support for economical, technical and project feasibility;
- Kick off: the actual development of the ontology begins with the ontology requirements specification;
- Refinement: in this phase, the goal is to produce the ontology according to the specification given by the kick off phase. It includes the knowledge elicitation and formalization processes;
- Evaluation: the usefulness of the developed ontologies is evaluated;
- Application and Evolution: in this phase, rules and strategies for ontology maintenance have to be addressed.

Ontology Development 101 (McGuinness & Noy, 2001) is a sequential process for designing ontologies. It is partially inspired by the literature on object-oriented design, and by the experience of the authors with ontology-editing environments such as Protégé (Protege, 2000), Ontolingua (Ontolingua, 1997) and Chimaera (Chimaera, 2000). The basic steps of this methodology are the following ones:

- Determine domain and scope of the ontology, using for example competency questions;
- Consider reusing existing ontologies;
- Enumerate important terms in the ontology;
- Define the classes and the class hierarchy;
- Define the properties of classes;
- Define the constraint (datatype, allowed values, etc.) on the properties;

- Create instances.

The methodology that we propose in this document (see Section 4) follows the basic steps described in (Roussey, 2005, see the beginning of this section) and shares some similarities with the methodologies for ontology development reported above. However, none of the methodologies presented above were ready to be used in APOSDLE:

- Uschold and King present a methodology which is purely manual. APOSDLE needs a modelling methodology supported by automatic or semi-automatic tools;
- TOVE and Ontology Development 101 use competency questions to define the scope of the application domain. We have followed a different approach and based the first step on our methodology on questionnaires, workshops and usage scenarios. This change was introduced because questionnaires, workshops and usage scenarios were more appropriate to capture the notion of domain that the Application Partners had at the beginning of the modelling process;
- All the methodologies described above, with the exception of METHONTOLOGY, are focused towards the creation of formal models and do not support an explicit phase where informal models are created;
- The methodologies described above do not address the problem of supporting informal modelling with appropriate tools. The description of models is usually created using word processors or paper documents. Within APOSDLE we have suggested Semantic MediaWiki as a collaborative tool for informal modelling;
- METHONTOLOGY suggests to carry out the three main activities in parallel. We felt that a methodology composed of sequential and well-defined steps was easier to follow and simpler to implement in the context of APOSDLE, where we can rely on possibly not enough skilled knowledge engineers.

The methodologies presented above cover also aspects not yet considered in the current version of the Integrated Modelling Methodology:

- the current version of the Integrated Modelling Methodology does not explicitly address maintenance issues; this is due to the fact that the first version of the Integrated Modelling Methodology is focused on the creation of the models. Therefore maintenance is out of the scope of this first version. We will address this problem in the second version and we will carefully evaluate if we need to develop ad hoc maintenance techniques or if some standard methodology can be included to support maintenance;
- the current version of the Integrated Modelling Methodology does not include an explicit documentation phase, even if the descriptions contained in the Wiki can be considered a sort of documentation of the modelling activities;

4 The Integrated Modelling Methodology

4.1 Overview of the Integrated Modelling Methodology

The methodology that we propose for building the domain-dependent models of APOSDLE Knowledge Base consists of five phases to be performed sequentially:

Phase 0. Scope & Boundaries: the scope and boundaries of the application domain are determined and documented;

Phase 1. Knowledge Acquisition: knowledge is (i) elicited from domain experts (using techniques like interviews, card sorting and laddering) and (ii) extracted from available digital resources relevant for the domain (using tools like a terms extractor);

Phase 2. Informal Modelling: an informal but rather complete description of the models is created using a Semantic MediaWiki;

Phase 3. Formal Modelling and Integration: each formal model is built using the appropriate formal language;

Phase 4. Validation & Revision: the formal models are evaluated and, if necessary, revised accordingly.

Phases 0—4 overlap almost exactly with the first five steps identified by (Roussey, 2005) and described in the previous Section. In the remaining of this section (Section 4.2 to Section 4.6) we provide a detailed description of each phase of the Integrated Modelling Methodology[3].

4.1.1 Main roles

Below one can find the definition of the main roles used in the Integrated Modelling Methodology, together with shorthand or acronyms used in the description.

- **Domain Expert** (DE): The DE provides the fundamental knowledge about the domain of the users of APOSDLE and their learning needs. The DE also specifies the pool of resources to be used for knowledge extraction.
- **Knowledge Engineer** (KE): The KE helps the elicitation of knowledge from the DE and guides the entire modelling process.
- **Coach**: The coach is a person who comes from the APOSDLE team and has the task of supporting Knowledge Engineers, who are not completely skilled in modelling, throughout the entire modelling process.

4.1.2 The Knowledge Bases of the 2nd Prototype

We briefly summarise the four application domains chosen by the different Application Partners to be part of the ASPODLE 2nd Prototype. A full description of these domains and the models produced is given in Deliverable D6.8 – Application Partner Domain Models. We also provide an overview of the team who performed the modelling for each Application Partner:

[3] An overview of Phase 1 and Phase 2 of the Integrated Modelling Methodology can also be found in Deliverable D6.8 - Application Partner Domain Models.

- Within ISN the focus is on project processing and management in order to provide specialized know-how in the area of innovation and knowledge management to their customer organizations. The modelling task was done by domain experts and by knowledge engineers from ISN, plus a coach from the APOSDLE team.
- The target group at CNM is the consulting department responsible for the Requirement Engineer process. After some initial thinking and exploring different possibilities, CNM has decided not to focus on a specific application domain but to further develop the RESCUE domain, a methodology for Requirement Engineering developed by City University and used as the application domain of the 1st ASPOSDLE Prototype. Therefore, the modelling tasks at CNM were performed with the additional help of experts from City University, plus coaches from the APOSDLE team.
- CCI Darmstadt needs continuous and permanent learning and training of all its employees about general information consultancy processes and topics focusing on REACH, a topic from the environmental consultancy area. The modelling task was done by domain experts and by knowledge engineers from CCI, plus a coach from the APOSDLE team.
- Finally, EADS IW wants to implement the APOSDLE system for the Simulation Domain. This is due to the growing importance of simulations and the necessity to have faster operational performance of engineers in this domain. The modelling task was done by domain experts and by knowledge engineers from EADS, without the help of coaches from the APOSDLE team.

4.2 Phase 0: Scope & Boundaries

4.2.1 Goal

In this initial step of the methodology, the goal is to define the scope and boundaries of the respective application domain and to identify potential learning resources. The desired output of Phase 0 is threefold: A first, preliminary list of tasks (called process scribble) roughly specifies the tasks that have to be performed by workers in the application domain. Further, a first list of learning goals describes abilities, skills, and knowledge that should be present in people performing these tasks. Moreover, a collection of representative documents should indicate relevant learning resources. Scope & Boundaries for the domain that are identified in this modelling phase have to be documented in an appropriate manner.

4.2.2 Defining the Scope & Boundaries and Collecting Resources for Prototype 2

Basically, Phase 0 was subdivided into three parts. At the beginning, an initial questionnaire on target groups, target tasks, and learning needs of the application domain was filled in by the KE in cooperation with the involved parties in the company (DEs, future learners, decision-makers in the company). In a workshop, the KEs were informed about the the procedure of modelling (this methodology) and the roles of models in APOSDLE. Candidate application domains were discussed, and the final domain was decided upon. By means of concrete, written scenarios, tasks were identified and additional learning needs were derived. Simultaneously, resources were collected that constituted potential learning materials. The outcome of this phase was documented in a specifically created modelling wiki. The wiki was chosen as a state of the art tool able to support the collaborative development of documentation concerning the models.

4.2.2.1 Questionnaire

As mentioned above, an initial questionnaire was used to gain insight about properties of the application domain. The questions were about target groups and tasks, the application domain, typical learning needs and insufficient learning support, high level learning goals (central domain concepts), experts and bottlenecks experienced during learning from experts, and about existing sources. The answers were collected and refined in written form in a specifically created Wiki and served as input for subsequent discussions, and for the final selection of the application domain. The questionnaire used for Prototype 2 is included in Appendix 1. The questionnaires with the answers provided by the different Application Partners are available from the modelling wiki (https://aposdle.fbk.eu/modelling/).

4.2.2.2 Workshop

For Prototype 2, a workshop was held by the coaches for the KEs of the different organizations. During this workshop task, domain and learning goal models in general, and their usage in APOSDLE were explained. The questionnaire (see Section 4.2.2.1) was discussed for each application partner, especially with regard to the choice for one application domain. KEs started to draft their task and domain models graphically, as an exercise to get a first impression about modelling. An overview over the procedure of modelling (this methodology) was given to the KEs.

Note that most KEs had not done task or domain modelling before.

4.2.2.3 Usage Scenarios

For further refining the scope of the learning domain, the KEs were asked to provide some usage scenarios that were describing concrete examples for work situations in which learning typically occurred. The KE should think of a concrete or prototypical person within the company, and describe a typical workplace-learning situation. That way, typical work-related questions in the respective situations were made explicit. Those scenarios were leading to a description of typical learning goals, and relevant domain concepts. As the outcome of the questionnaire, the usage scenarios were recorded in written form in the modelling wiki.

4.2.2.4 Resources

Resources and relevant documents have to be collected throughout the entire modelling process. This ensures that the models are in line with the resources available. In the Scope & Boundaries step, the KE was collecting several documents which were being used in the next step of the methodology for eliciting knowledge (term extraction, card sorting). Therefore, these resources had to be locally available for the KE..

4.2.3 Remarks about this Phase

The implementation of Phase 0 for Prototype 2, and the feedback obtained from Application Partners and coaches, has shown that it provides good support towards the achievement of its goals and objectives. An important aspect of this Phase, which we are not able to support at the current stage, that is, the development of guidelines and criteria to decide on appropriate learning domains for APOSDLE among different possible candidate learning domains. To provide such guidelines and criteria, and to decide whether APSODLE can be meaningfully used for any learning domain or is better tailored to application domains with certain characteristics, is a difficult task which involves not only the group working on the Integrated Modelling Methodology but different parts of the consortium and should be based on input received from the evaluation of Prototype 2 contained in Deliverable D6.15. Starting from the evaluation of Prototype 2, we will investigate if we have enough material to compose some guideline and start to address this problem for future prototypes. Another aspect we realised while applying the Integrated Modelling Methodology is that while certain application partners have both the task model and the domain model in the same application domain, others (in our case ISN) have general descriptions of processes, for instance the consultancy process, which can be applied to several different domain models. This latter case was quite challenging and we need to better understand if APOSDLE is the appropriate tool to deal with similar application domains.

4.3 Phase 1: Knowledge Acquisition

4.3.1 Goal

The goal of the Knowledge Acquisition step is to extract as much knowledge as possible from both, digital resources provided by the DEs, and by eliciting knowledge directly from the DEs. The results are a refined task list and an extensive list of candidate domain concepts that are documented in the modelling wiki.

4.3.2 Knowledge Acquisition for Prototype 2

Phase 1 is subdivided into two different activities, namely "Knowledge Acquisition from Digital Sources", and "Knowledge Elicitation from Domain Experts". The two activities are running in parallel.

4.3.2.1 Knowledge Acquisition from Digital Sources

From the digital resources that were collected in Step 0, the KE extracts terms using terms extraction tools. The term extraction tool used for extracting terms from documents for Prototype 2 was the Domain Modelling Tool (DMT) described in (Pammer, Scheir, & Lindstaedt, 2007). The Domain Modelling Tool consists of two plug-ins for Protégé OWL. The first, Discovery Tab, supports document-based ontology engineering with relevant term extraction, clustering and related functionalities. The second, Annotation Tab, provides a facility to annotate documents manually and automatically (based on a training set). For knowledge acquisition, only the Discovery Tab was used. The Discovery Tab lets the user extract relevant terms from sets of documents and creates ontology classes and properties from these terms. The version of the DMT (Version 1.1) that was used in this phase for Prototype 2 provides basic support for German language and allows for loading externally created customized stopword lists.

The output of the automatic term extraction is a flat list of terms, which must be checked and refined by the KE during the Informal Modelling Phase (Phase 2).

4.3.2.2 Knowledge Elicitation from Domain Experts

In order to elicit knowledge from the DEs, the KE applied various techniques. Structured interviews were conducted by the KE for refining the list of domain concepts, the list of tasks, the relations between tasks and the mappings between tasks and domain concepts. Special methods (card sorting, laddering) were applied for eliciting tacit knowledge from the DEs.

4.3.2.2.1 Structured Interviews

Based on the first-cut task list and process scribble that was generated in Phase 0, a more fine-grained task list should be generated, and further relevant domain concepts should be identified. Therefore, the KE conducted structured interviews with the DE. Tasks were broken down into sub-tasks by asking for each task in the first-cut task list, what were its subtasks. This was performed, until the KE and DE were able to obtain the desired granularity of the task list. The decision whether the task list is fine-grained enough depends on the intended use of the learning environment, and on the intended target group, and is hence on the knowledge engineer. As a rough guideline, the task should allow for a manageable amount of learning goals that can be acquired in a reasonable time by the intended target group, to allow for work-integrated learning. However, defining objective criteria for the ideal degree of granularity of the task list is still an open issue, and one of our research questions in future work.

Relations between tasks were identified by asking for each task and each sub-task, what input would be required, and what the output was. Relevant domain concepts and learning goals for a task or a sub-task should be elicited by asking for each task, what knowledge was needed for accomplishing the task, and for guidelines on performing the task.

4.3.2.2.2 Card Sorting

Card Sorting is a technique that is very often applied in information design processes in order to generate an overall structure for information, as well as suggestions for navigation, menus, and possible taxonomies. In Phase 1 of the modelling methodology, it was performed with the DEs to find relations between domain concepts, and for identifying new relevant domain concepts. Card sorting is a quick, inexpensive, and reliable method that can provide insight to the DE's mental models and that can make tacit knowledge explicit.

For eliciting expert knowledge for Prototype 2, Card Sorting was performed as follows. The KE, with the help of coaches if applicable, prepared a set of cards (objects) with a clear description for each one of them. The set of objects could be a set of resources or a set of previously chosen domain/task concepts. The KE mixed the cards and gave them to the DEs, asking the DEs to sort the cards in different sets. The KE then asked the DEs to specify the criterion used to sort the cards. Better results can be usually obtained by small groups of DEs working together. The KE documented the piles obtained and the sorting criterion applied by the DEs. Then, the KE mixed the cards again and gave them back to the DEs asking for a new sort according to a different criterion. Sorting with the same set of cards should proceed as long as the DEs are still able to come up with different sorting criteria.

4.3.2.2.3 Laddering

Laddering is a semi-structured interview technique that is employed in order to break-down and refine identified domain concepts, or to detect relationships between concepts. The KE starts from one domain concept, for example, the high level concept “Method”. For instance, the question “Which methods do exist?” leads to a number of domain concepts, for instance, “Brainstorming Techniques”, or “Knowledge Management Techniques” that are connected to the starting concept by a relation of type “is-a”. This procedure is then repeated for the new concept, for example. by asking “Which Knowledge Management Techniques do exist?”, and also for the resulting domain concepts until the desired degree of granularity is obtained. In so doing, a “cognitive ladder” between different domain concepts is established. As for the granularity of the task list, the decision about the granularity of domain concepts is on the knowledge engineer. For taking this decision, several factors, such as the resources available, but also the documents to be created in the future, or the potential learning goals for workers have to be taken into account.

4.3.3 Remarks about this Phase

The objectives and methodology developed for Phase 1 provided good results but can be improved in several ways, especially regarding Knowledge Acquisition from Digital Sources. The term extraction feature implemented in the DMT (Version 1.1) provided only some basic support for the German language – used by most Application Partners – and is judged to be not fully satisfactory in several respects by several Application Partners and Coaches (see feedback questionnaires in Appendix 2). This basic support could be improved, leading to more accurate suggestions of candidate concepts made by the DMT. Furthermore, the term extraction algorithms need to be designed in the direction of having a relatively small (compared to web-size dimensions of millions of pages) number of potentially large size resources (for example, in the case of manuals). Currently, statistical methods relying on term frequencies and their frequencies in multiple documents are used (TF-IDF) for determining term relevancy. These methods perform naturally better in large corpora (especially more documents) where the influence of outliers is smaller. In the future we plan to improve the underlying algorithms, and also add better support for the German language.

Another resource which could be used to discover candidate concepts and basic taxonomies, is semi-structured information already present in the different organizations: directory structures, database schemata, or content repository managers which can provide useful insights of the knowledge contained in the application domain. One of the possibilities for future enhancements is to explore if, and how, semi-structured information can be used to elicit knowledge from the application domain resources directly.

4.4 Phase 2: Informal Modelling

4.4.1 Goal

Starting from the knowledge elicited in Phase 1 (see 4.3), the main goal of this step is to obtain an informal but complete description of the domain and the task model in a Semantic Wiki[4]. After this modelling step, the informal models should only consist of relevant domain concepts and task descriptions (see Section 4.4.3).

4.4.2 General

The goal of Phase 1 was to acquire as much information as possible. In this modelling stage, the KEs has to review and filter all the candidate model elements and to retain only relevant domain concepts and tasks. These relevant domain concepts and tasks have to be documented in an informal but rather complete manner.

The tool we use to support the informal modelling phase is a Semantic MediaWiki. The reason to use a Semantic Wiki is that it allows the KEs to provide the descriptions of the elements of the task and domain model in Natural Language. However, at the same time it allows to structure these description according to pre-defined templates and allows to use semantic constructs like attributes and relations, so that the informal descriptions in Natural Language contain enough structure to be automatically translated in formal models (e.g., in OWL ontologies), without forcing the Application Partners to become experts in the formal languages OWL and YAWL.

The main idea is to let the KE fill two types of template, one for each concept of the domain model and one for each task of the task model. It is possible to fill the templates in the language of the Application Partner (in our case, French or German) but additionally an English translation and an English description has to be provided for both tasks and domain concepts in order to allow also non German or French technical partners to contribute to the definition of the models.

The output of this step is a list of filled templates, one for each task and one for each domain concept.

4.4.3 Informal Domain Model

Starting from all possibly relevant terms generated in Phase 1 (see 4.2), the KE decides which of them are relevant domain concepts, and which should be discarded, and prepares a list to be validated by the DEs. To help deciding about relevant concepts in the APOSDLE knowledge base, the following guidelines were given:

1. Is this domain concept useful for retrieval?
    1.1. Are there resources dealing with this domain concept, or is it reasonable to expect resources dealing with this domain concept in the future?
    1.2. Does this domain concept help to differentiate between resources?

---

[4] To support our methodology we have chosen Semantic MediaWiki.

2. Does this domain concept refer to a learning goal of a hypothetical APOSDLE user?
   2.1. Does this concept help APOSDLE to support the mastering of the learning goal?

As a general rule, it is suggested to keep possibly "irrelevant" concepts rather than risking to remove relevant ones. Nevertheless it is clear that the consequence of keeping irrelevant information at this stage increases the effort of modelling in every subsequent modelling stage.

4.4.3.1 Domain concept template

Figure 4-1 shows a screenshot of the domain concept template in the Semantic MediaWiki.

4.4.3.1.1 Rationale

For each concept we ask for "*Name*", "*English name*", "*Short description*", "*English description*" and "*Synonyms*". These elements are modelled as String attributes in the Semantic MediaWiki. We also ask for some relation with other concepts, suggesting pre-defined relations such as "*Is a*", "*Is part of*", and "*Is related to*". We also ask the users to instantiate the generic "*is related to*" to describe domain dependent relations, which later can be made more specific. All these relations are modelled as relations also in the Semantic MediaWiki. From a technical point of view the difference between attributes and relations is that attributes point to vales (String values in these cases) whereas relations point to other Wiki pages.

The predefined relation "*Is a*" is introduced in the template with the subclass-relationship of OWL in mind. Therefore the informal "Is a" relation in the Semantic MediaWiki is used with the semantics of the subclass-relationship of OWL (For two concepts X and Y, "X Is a Y" if everything that is an X, is also a Y) and is also automatically transformed in the is-a subclass-relationship of OWL when going from informal to formal (see Section 4.5).

4.4.3.1.2 Usage

The KE starts filling the template, providing information for all the attributes and relations. At first the KE can use the generic relation between concepts "*Is related to*" to link concepts. In the course of creating the informal domain model however, the KE can create specific, domain dependent, relations replacing "*Is related to*" with more appropriate relations. To provide a simple example, if the concept "*Sweater*" and the concept "*Wool*" are, in a first round, linked with the generic relation "*Is related to*", this relation can be replaced with a more appropriate relation "*Is made of*".

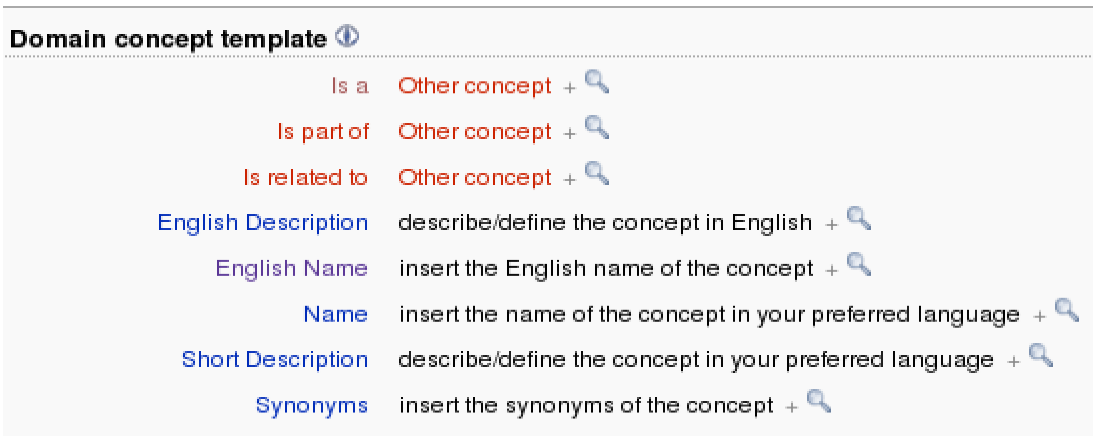

Figure 4-1: Screenshot of the domain concept template in the Semantic MediaWiki

4.4.3.2 Properties of a completed informal domain model

At the end of the informal modelling phase, the informal domain model should be checked for two different aspects, a technical and a conceptual aspect.

- **Technical aspect**: Each domain concept should be described by a complete domain concept template. A complete domain concept template has a values for each of the attributes ”*Name*”, ”*English name*”, ”*Short description*”, ”*English description*” and ”*Synonyms*”. In addition information concerning "*Is a*", "*Is part of*" should be provided and should connect the concept with other domain concepts (whose templates also need to be complete). The generic relation "*Is related to*" should be replaced by domain specific relations.

- **Conceptual aspect**: All domain concepts should be relevant and unambiguous, and all created relations should also be unambiguous. By unambiguous we mean that a concept or a relation should only have one meaning. Typically, general concepts/relations are in danger of being ambiguous (for example, "resource" or "contains"). Clear, intelligible verbal descriptions usually help to disambiguate them.

4.4.4 Informal Task Model

Starting from all possibly relevant tasks generated Phase 1 (see 4.2), the KE decides which of them are relevant tasks and which should be discarded. To help deciding about relevant tasks in the APOSDLE knowledge base, the following guidelines were given:

1. Does the task refers to a situation / task in which learning supported by APOSDLE shall occur?
   1.1. Does the task require knowledge that lies inside the specified learning domain?
   1.2. Should APOSDLE be able to support this task?
2. Is the task recognisable for the future APOSDLE user?
   2.1. The statement “I am currently doing task X” should make sense to the future APOSDLE user and should provide a good insight in the correct granularity level of a task. For instance, the statement “I am currently performing an activity” would be too generic, and it probably would not make sense to a user, while the statement “I’m currently pressing the “ESC” character” would probably be too specific and in most cases would not make sense for a user either.

4.4.4.1 Task template

Figure 4-2 shows a screenshot of the task template in the Semantic MediaWiki

4.4.4.1.1 Rationale

For each task we ask for “*Task title*“, “*English name*“, “*Task description*“, “*English task description*“. These elements are modelled as String attributes in the Semantic MediaWiki. “*After*”, “*Alternative to*”, “*Before*”, “*Has supertask*”, “*Knowledge required*”, “*Parallel*”, ”*Task type*” are relations in the Semantic MediaWiki and have been defined with the semantics of YAWL in mind. The “*Has supertask*” relation has the following semantics: for tasks X and Y “X Has supertask Y” means that X is a more fine-grained task that is part of Y.

The predefined “*Knowledge required*” relation has been defined for anticipating the more complex Task – Learning Goal – Domain Concept mapping. Doing this, we want to capture early on in the modelling process the relation between tasks and domain concepts. The semantics is as follows: For task X and domain concept Y “X Knowledge required Y” means that in order to successfully perform X, knowledge about the concept Y is necessary. This relation allows the coherent definition of task, learning goals and domain model, as specified by the APOSDLE-inherent requirements in Section 2.

There is no formal semantics for this relation however, as this relation will need to be re-examined and formalised in Phase 3 (see Section 4.5).

The predefined relation "*Task type*" points to one of a number of predefined task types in a Task Catalogue (Wiki pages in the Semantic MediaWiki). The explanation of task types is out of the scope of this deliverable. In a nutshell they have been introduced in APOSDLE in order to be able to categorise tasks on a more general level so that this information can be used to tailor the learning suggestions.

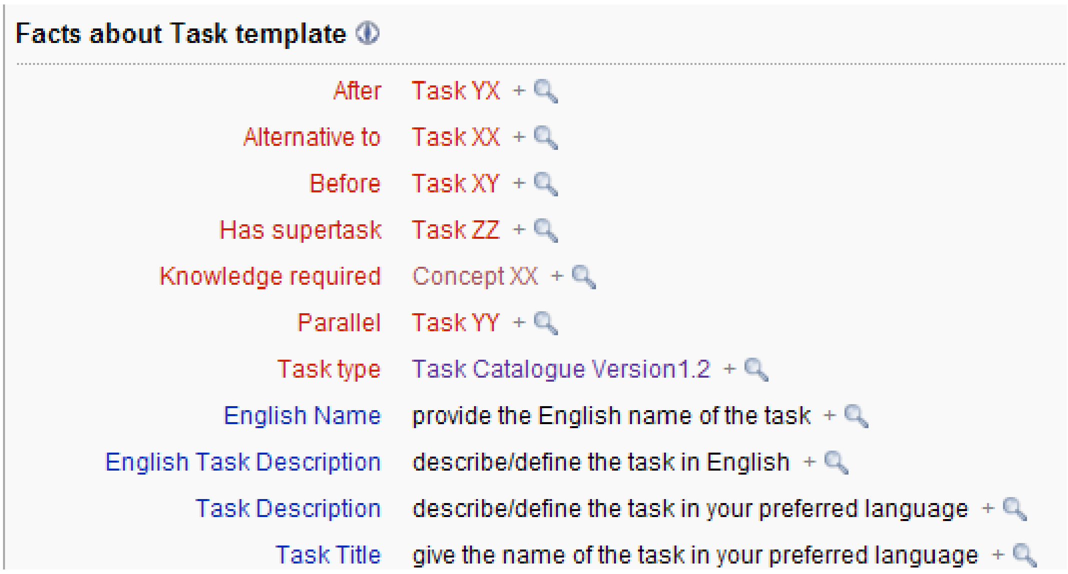

Figure 4-2 Screenshot of the task template in the Semantic MediaWiki

4.4.4.1.2 Usage

The "*Knowledge required*" field in the task template is optional. If the KE wants to specify knowledge about relevant domain elements already in this modelling phase, (s)he can do this here. The "*Knowledge required*" has to be specified in terms of one or more domain concepts. This is potentially a good moment to discover relevant domain concepts missing in the informal domain model.

The "*Knowledge required*" section does not have to be a complete list of domain concepts that are relevant for performing the task. It should be regarded as a possibility to record task - domain concept-mappings that will be taken into account in Phase 3.

A task's required knowledge should be specified at the lowest level of the task-subtask hierarchy, that is, if a task has sub-tasks, then the required knowledge should be specified only for the subtasks.

4.4.4.2 Properties of a finished informal task model

The informal task model should be checked for two different aspects, a technical and a conceptual aspect.

- **Technical aspect**: All tasks should be described by complete task templates. A complete task template has values for each of the attributes and the relations "After", "Alternative to", "Before", "Has supertask", "Parallel", point only to other tasks (whose templates also need to be complete). The relation "Task type" is filled and refers to one of the task types in the Task Catalogue. If the "Knowledge required" relation is filled, it refers to domain concepts (whose templates also need to be complete).

- **Conceptual aspect**: All tasks should be relevant and clearly defined, that is, have a clear and intelligible verbal description.

4.4.5 Prototype 2

Initially, templates for task and domain concepts were provided in a (non-semantic) MediaWiki. While the informal modelling was still going on, the contents of the MediaWiki were migrated to the Semantic MediaWiki. The KEs of each application partner had to process each task and domain concept template in order to add the semantics missing in the first version.

To explain with an example, if in the MediaWiki the relation was:

English name Sweater

in the Semantic MediaWiki the KE had to write:

[[English name:=Sweater]]

Furthermore, since a single Semantic MediaWiki for all partners was set-up, each task or domain concept had to be assigned to a category identifying the AP and whether the page is a task or a domain concept, like (EADS-task, ISN-concept and so on).

Both these changes increased the workload of the KEs slightly during the preparation of informal models for Prototype 2, compared to what would be the normal course of modelling. Nevertheless, the introduction of the semantic media wiki (instead of a normal wiki) has allowed to make the following steps concerning the production of the formal models (done automatically for the domain model) and the integration of models (due to some knowledge extracted from the semantic wiki and given in input to the TACT tool – see Section 4.5.2.3), less time consuming and labour intensive.

4.4.6 Remarks about this Phase

The methodology and tools developed for Phase 2, based on the usage of a Semantic Wiki, and templates of description has provided, in most cases, good quality informal models and has allowed a precise description of the different elements (tasks, domain concepts) of the models of the APOSDLE knowledge base. For two of the four application domains, tasks and domain concepts were well documented by the APs, and additional information such as the "Knowledge Required" by a task and the "Task Type" required by the TACT tool were already provided in early phases of the modelling, allowing an integrated development of the different models. Nevertheless, the usage of the Semantic MediaWiki as such resulted in tedious and excessively time consuming work from the KEs, as Semantic MediaWiki is not designed to be a modelling tool. This produced some delay and incompleteness in the informal models of the remaining two application domains, which had to be fixed in the phase of formal modelling and integration. Also it has highlighted several improvements that need to be added to the Semantic MediaWiki in order to be used as an effective tool for informal modelling. For future prototypes we plan to improve the Semantic MediaWiki extensively in order to make the task of informal modelling less tedious and time consuming. Most of these improvements are discussed and described in Section 5.4.

4.5 Phase 3: Formal Modelling and Integration

4.5.1 Goals of the phase

Starting from the informal modelling done in phase 2, we create the formal models which constitute the APOSDLE knowledge base, together with the relations connecting them, in a way which is to a large extent automated.

The steps performed to produce formal models differ for the domain model and for the task model. They are both based on the outcome of the informal modelling phase, that is, on the filled templates of tasks and domain concepts. In a nutshell, the outcome of the informal modelling phase is transformed, which also becomes explicit by a shift of tools: the content of the Semantic MediaWiki is transferred into models in OWL (for the domain) and YAWL (for the tasks) respectively.

We proceed in the following order: first we will explain the necessary steps for the task model, then we will do the same for the domain model, next we present the generation of learning goals based on a "typed" mapping of task model and domain model.

4.5.2 Formal Modelling Steps

4.5.2.1 Task model

The task model was manually defined in YAWL starting from the informal descriptions provided in the Semantic Wiki. The strategy we have applied for the formal modelling of the tasks is based on the work of Reijers (Reijers, 2005). This work focuses mainly on the concept of process "re-design" and surveyed methodologies of task modelling, which originated in consulting agencies work in most of the cases. Reijers assumes, that formal task modelling is almost never in the position to define a work process from scratch. However, it may be a chance to restructure and optimize work. One branch of the methodological work presented in (Reijers, 2003) structures human activity monitoring in such a way, that alternative formal models expressed as Petri Nets (van der Aalst, 1998) are derived and graphically presented to the domain experts. The necessary human activity modelling includes intra-organisational data flows (e.g. filling out particular document forms and passing them to a particular owner) as well as probabilities of sequence, parallelism and combinations of actions, which trigger the intra-organisational data flows. As such information is not included in our informal models and even not intended by the APOSDLE system, which uses a task model more in the sense of an intelligent way of presenting learning content, we abolish this so-called "information-based" approach. We prefer to base our methodology on the best practice approach of (Reijers, 2005). The model fully abstracts from any data flow. Note that this kind of abstraction is also the reason why we only apply Reijer's best practices for control flows (Reijers, 2003).

We proceed under the assumption that a collection of the tasks already exists – at least as a list. The informal task model along with the templates presented in the previous section matches this requirement. Moreover, the informal task model provides information on the parallelization, elimination, sequencing and splitting of tasks. These are the best practices to be applied to the informal model and to be translated to YAWL.

Due to the way the informal models are created, the application of the best practices comes down to a check. The respective language constructs are:

- "AND" for parallelization. We do not distinguish between strict parallelism and interleaved tasks, as there is no data flow. The reason is again the full abstraction from any data flow. For instance, tasks like "search information for consulting" and "structure information for consulting" would actually include passing data (e.g. the search results from a search engine are written down in an MS Word template). Nevertheless we abstract from that and model "search information for consulting" and "structure information for consulting" as parallel tasks.
- The modelling of alternatives must be expressed by the lazy "OR" construct explained in (Godehardt et al., 2007). Because of the application logic of the YAWL workflow engine, applying the basic "OR" provided by the YAWL editor would force the user to chose all alternatives of a working step in advance. The lazy "OR" enables extending or restriction the initial choice.
- "XOR" between a task and an unnamed dummy task for leaving the step out creates optional tasks. For instance, modelling an "XOR" between "structure information for consulting" and an

unnamed task would make "structure information for consulting" optional and assign the ability to skip this step by the unnamed "empty" task.

- The regular YAWL sequencing arrow for sequences.

The final step for the task model is the application of formal checks provided by the YAWL editor. The specification can be checked regarding the connectedness of sequences, its ability to terminate in general and the behaviour of loops in particular.

These checks result in a debugging of the YAWL specification. This outcome is complemented by a transformation of the YAWL specification to an OWL file containing the complete list of tasks.

4.5.2.2 Domain model

Technically speaking, the RDF exported in the informal modelling step is transformed into an OWL model. First, the RDF is imported into the Protégé editor. Complete lists of relevant terms (nouns, verbs, phrases) have either already been produced in the informal modelling phase or can be additionally extracted, clustered and shaped into hierarchies from documents relevant to the domain. The tool provided by APOSDLE is a term extraction and clustering tab extension of Protégé (Protégé, 2000). Protégé is also applied to identify concepts (nouns our noun-like phrases) with *is-a* hierarchy identify relations (verbs, verb-like phrases).

Similar to the formal task modelling step, the work to be done ranges from defining the hierarchy and the relations from scratch to re-using the respective wiki information. Ideally, the activities can be narrowed down to checks of the hierarchy, the granularity of the concepts and the relations. Useful questions to be resolved by the application of Protégé are:

- Does each *is-a* relation hold? The most straightforward procedure to determine this is browsing the hierarchy in Protégé bottom-up or top-down.
- Is the degree of specialization of the hierarchy sufficient and uniform? A method to determine this is browsing the most special concepts in Protégé, that are the concepts without any sub-concepts. Similarly, it proves to be helpful to browse the concepts, which are on the same level of abstraction, as, for example, the same number of *is-a*-steps away from the root concept.
- Do the concepts match the extracted terms from the extractor and thus the learning material?

As a rule of thumb for cases where these three procedures suggest the extension of the domain model, the concepts should not be overspecialized, and possible further specialisations left for future extensions of the models, only if required.

The creation of instances and the addition of axioms in Protégé, statements about individuals and restrictions on concepts can be added if needed. This would be truly additional work, as the informal modelling phase does not provide information on instances and axioms in the sense of OWL.

The final step for the domain model is the application of formal checks provided by Protégé to determine (in)consistencies in the resulting ontology.

4.5.2.3 Learning goal model

We regard a learning goal as the combination of a "Learning goal Type" and a "Domain Concept". The domain concept defines the content or topic that the learning goal is about. Learning goal types specify the type and degree of knowledge and skills the person must or typically wishes acquire about this topic. An example of learning goal is "Apply METHONTOLOGY", where "Apply" is the learning goal type and "METHONTOLOGY" is the topic that the learning goal is about.

Before defining the learning goal model, we require that a formal task model and a formal domain model already exist. Additionally we use as a starting point a first rough list of mappings between tasks and domain concepts (knowledge required) already provided during the informal modelling step.

The list of mappings automatically extracted from the wiki is adjusted using the extensions of tasks and concepts found in the formal modelling steps for the task and domain model. The resulting adjusted list of mappings and the formal models of tasks and domain are the initial input for the TACT tool (TACT is the abbreviation for Task And Competency Tool). TACT facilitates the production and mapping of up to four different learning goals per concept *x* (that means for a concept *x* from the domain model: remember *x*, understand *x*, apply *x*, create *x*) and integrates the models by establishing links between the task-model and the domain model. This strategy was designed to avoid complex matrix expressions of the mappings, requiring a less adequate tool like Microsoft Excel. Instead, the initial mapping from the informal phase can be iterated. The learning goal types to be expressed by this linkage are based on the task types created in the informal modelling phase – consequently, TACT gives a suggestion about this too. The learning goal model produced by TACT, plus the final list of mappings between tasks and competences are directly stored in OWL ontologies, ready to be integrated in the APOSDLE knowledge base.

4.5.3 List of the tools used

- The YAWL editor (Yaw) used for defining the formal task model has a graphical interface, which allows for manipulating a YAWL model graphically by drag and drop. As indicated above, we only used the control flow elements of this editor. As the tool also provides an XML interface, the conversion and wrapping of a YAWL specification into OWL, needed to easily integrate the YAWL model in the APOSDLE knowledge base, is straightforward.
- Protégé (Protégé) is a very popular ontology editor, which was extended for the purposes of APOSDLE. Protégé resembles an environment for software development and gives systematic access to a conceptual hierarchy along with domain- and a range of non-hierarchical relations. The extension for APOSDLE extracts keywords from texts and has additional clustering capabilities, which can be used to establish conceptual hierarchies.
- TACT is designed to load OWL-Specifications of tasks and concepts and to let the user correct the mappings from the informal modelling phase. It also makes use of the informally added “knowledge required” to some tasks in the Semantic MediaWiki, and suggests using these concepts for learning goals. This is a way to ensure that no ideas, or knowledge, is lost “on the way” through the many stages of modelling.
- The YAWL-XML and the RDF extracted from the Semantic Media Wiki are transferred to OWL by backend procedures. This is fully automated by Wiki2OWL converters created ad hoc for ASPODLE (see Deliverable D4.3 – Software Architecture for 2nd Prototype)

4.5.4 Remarks about this Phase

The transformation of the Semantic Wiki content to YAWL could be automated, as the YAWL tools provide the necessary XML interfaces. Nevertheless, the introduction of a graphical view (YAWL editor) on the task model only after the translation from the Semantic Wiki to YAWL seems not very effective, at least in cases, where the knowledge engineer already starts with some kind of organisational task model. This means, that at least for the task model, the order of steps in the initial phase, the informal phase and the formal phase should be reworked, and some graphical support provided also for the informal phase. This becomes even more interesting under the assumption that the automatic task detection in APOSDLE may provide useful information on how the user interacts with the task model.

4.6 Phase 4: Validation & Revision

4.6.1 Goals of the phase

The integrated model resulting from Phase 3 is a unique complex knowledge representation, mapping very different components: a process definition (task model), a knowledge space (Korossy, 1997) similar to a formal concept lattice (learning goal model), and an ontology (domain model).

Consequently, the goal of the “Validation & Revision” step is to verify the correctness and completeness of all these components that compose the integrated formal model.

4.6.2 Model validation and revision for Prototype 2

The model’s validation consists of checking separately each of the formal models (task and domain) and their integrated version. The checks are performed by querying the APOSDLE Knowledge Base and by manually evaluating the results. The language used to perform the queries is the query language for RDF SPARQL .

The first group of checks is dedicated to evaluate the relations existing between the different models. We determine the list of tasks that haven't got any associated learning goal, as in a running APOSDLE application this would not trigger learning-goal based searches, which would limit functionality. Tasks with no associated learning goals are likely to trigger a revision of the models, as they are likely to be “errors” of the modelling. Nevertheless this need to be evaluated by the KE, as the designers of the formal models might decide to leave some abstract tasks, which merely serve for navigation purposes or to represent only the knowledge worker real work context, not associated with learning goals.

In a similar way, our check for tasks that are associated with more then five learning goals should prevent an overly long list of search results in the running APOSDLE system. Tasks with more than five learning goals may show some granularity problems, and may suggest to split tasks in more fine grained ones, Again the knowledge engineer might decide that linking a task to many different learning goals is necessary and may not revise the models.

In order to support the KE in checking the tasks’ relevance, the tasks that have exactly the same learning goals are also identified. This could suggest to join, or to delete, tasks with this property.

Starting from the domain model, an additional test aims at selecting the domain elements that are not connected to any task. There are several possible reasons for a domain concept having no task attached. The learning goal model might be incomplete or the domain model might contain a number of concepts that are not relevant for learning. In the first case, the learning goal model has to be revised. Regarding the second situation, the KE should reflect on the domain model and decide if a given concept is relevant and useful for improving information retrieval (it might be task independent) or if the concept is “irrelevant” and should be removed.

Finally, there is also a need to check if the domain concept, manually associated to the learning goals, belongs to the domain ontology. In other words, we need to make sure that the required knowledge elements, associated to each task during the informal modelling phase and used by the TACT tool to suggest competences, are part of the domain ontology. As indicated in Section 4.4.4.1, the “knowledge required” field in the task template is optional and it gives only the opportunity to specify some knowledge about relevant domain elements. Thus, the user can specify whether she wants to create a mapping between a task and new knowledge element that does not belong to the domain concept.

The second group of checks prunes the single models. We apply checks for the first-level concepts that have no sub-concepts (isolated concepts) and for tasks that haven't got any associated task type. As for the isolated concepts, the knowledge engineer should decide, if these can be discarded or should be kept for extendibility purposes. Similarly, for tasks without a task type. Additionally, based on the list of learning goal types associated to each concept, the KE evaluates the learning goal model. Checks for the task types and learning goal types which are never used, complete the test suite.

4.6.3 Remarks about this Phase

The use of SPARQL queries allows checking typical properties, characteristics, or possible anomalies of the APOSDLE knowledge base, which are issues for the Knowledge Engineer. An additional source of information for the validation phase would be to bring the Domain Experts in the picture, asking them if the Knowledge Base produced satisfies their initial description of the application domain. This poses a problem, as the Domain Experts often do not know (and have no time to become acquainted with) formal languages like OWL or YAWL. An idea is to extend the validation of the integrated formal models using verbalization tools which allow to translate formally defined models into natural language. This would allow the Domain Experts, who may be not familiar with formal languages, to validate models.

Including verbalization tools in our methodology needs to be considered carefully, and is not an easy task, as verbalization of ontologies is an active field of research, with several very active projects, but there are still no mature tools yet. An example of ongoing research projects and initiatives that aim at providing more natural language representations of ontologies are:

- Verbalization of OWL ontologies on the basis of predefined templates (Jarrar, Keet, & Dongilli, 2006);
- Verbalization of OWL in controlled English (Kaljurand & Fuchs, 2007);
- Generation of the NL paraphrases for OWL concepts based on variety of NLP techniques and implemented in SWOOP (Hewlett et al., 2005).

In order to provide the most appropriate verbalization tool or technique, we have to take into consideration the peculiar nature of the APOSDLE models. A first important aspect is multilingualism: most of the APOSDLE knowledge base models are developed in French and German, while most of the current research is devoted to the translation of OWL to English. Another aspect is the verbalization of YAWL, for which there is not much active research. Therefore, while we would like to include some of these ideas and tools in the methodology, the feasibility of this is still an open issue.

5 Evaluation & Further Development

In this Section we report some feedback and comments by the Application Partners and the Coaches on the modelling methodology proposed. We provide a first evaluation of the methodology and ideas for further development. The feedback has been collected from the following sources:

1. APOSDLE Deliverable D6.8 – Application Partner Domain Models;
2. Discussion at the Plenary & General Assembly Meeting, Paris, 10-12 October 2007;
3. Feedback Questionnaires. The questionnaires were sent to the Application Partners and their Coaches and were composed of specific questions for each step of the methodology. The filled feedback questionnaires are in Appendix 2.

We organize the evaluation as follows: first, we consider some general comments on the entire methodology; then, for each phase of the methodology, we report the specific feedback for that phase. For each comment, we report if it was made by Application Partners (APs) or by Coaches. Furthermore, if a comment suggests some improvement or modification of the methodology, we propose one or more possible actions to tackle or solve the problem for future versions of the methodology. The feedback for the single phases concerns several different aspects: it goes from comments on general aspects to specific comments on technical problems. In order to help the reader we tag the comments for each phase with one of the following labels:

- [GENERAL] – we tag with this label those comments about a general aspect of a phase;
- [TOOLS] – we tag with this label those comments about tools and techniques proposed to support the methodology;
- [ORGANIZATION] - we tag with this label those comments about organizational aspects of the methodology.

Note that the feedback questionnaire stimulated the Application Partners and coaches to comment also on more general issues – not necessarily related to the methodology. In this section we process only the feedback that deals directly with the methodology.

5.1 General Feedback on the Methodology

From the experience of producing the APOSDLE knowledge bases for Prototype 2 we have identified the following general issues:

- **Granularity of models**. Guidelines for and examples of the granularity of the models are missing. Therefore it is difficult to understand the "right" granularity at which to perform modelling. (Source: APs)

  *Room for improvement***:** the granularity of models is a typical issue in modelling, and not only a problem of APOSDLE, for which there is no general solution. In order to provide some guidelines and examples, we think if two different actions: first we will study data from the evaluation of the four use cases of the APOSDLE 2nd Prototype to see if it is possible to extract some useful guidelines. Additionally, we have noticed that the granularity issue concerns mainly the domain model, and not the task modelling. This suggests a modelling strategy which could help addressing the granularity issue: first start with task modelling; second use the "knowledge required" attribute of tasks to build a "core" list of domain concepts; this core list of concepts should provide some suggestion on the granularity level required by the domain model.

- **Workload of modelling**. Too much workload/effort is required to follow the methodology and perform modeling. Furthermore, an estimate of the workload/effort for each phase is missing. (Source: APs)

  Room for improvement: For a system like APOSDLE, it is impossible to completely eliminate any modelling effort. However, we aim at reducing substantially the modelling effort by improving the tools which support the methodology, in particular the Domain Modelling Tool and the Semantic Wiki. Moreover, from the experience of producing the APOSDLE knowledge bases for Prototype 2, we are able to provide a better quantification of the effort required for each step of the methodology. This will help coaches and APs to better plan their work. In addition, all the effort spent for Prototype 2 to transform the content of the MediaWiki in a Semantic MediaWiki compliant form will not be required in the future as the latter will be used right from the start.

- **Importance of Knowledge Engineers.** The quality of the models created is much better when the APs can dedicate one or more persons to modelling activities. (Source: Coaches)

  Room for improvement: The methodology as it is now, relies on a committed Knowledge Engineer from the Application Partners. What to do if this person is not available is a general issue, common to many complex systems development projects that require some effort to be customised. A possible solution could be to provide external Knowledge Engineers, instead of coaches, but this is a decision which should be considered by the entire APOSDLE consortium and not only related to the Integrated Modelling Methodology.

- **Importance of coaches.** The role of the coach is really important in the methodology. Good support from coaches was fundamental to achieve good modelling modelling results (Source: APs and Coaches).

  Room for improvement: The methodology should provide guidelines and support also for coaches in order to improve the quality of coaching.

- **Importance of tools**. There is a close correlation between the quality of the models created and familiarity with, or a good understanding of, the tools used, in particular of the Semantic MediaWiki. (Source: Coaches)

  Room for improvement: The usability of tools is a fundamental issue that must be improved for future versions of the methodology. Additionally / alternatively, good user manuals should be provided / referenced together with the modelling methodology manual.

5.2 Feedback on Phase 0: Scope & Boundaries

5.2.1 Positive Feedback

- [GENERAL] This phase allows the KEs to acquire a better understanding of the domain to be modelled and to prepare the interviews with DEs. (Source: APs)
- [GENERAL] It is the most important step of the entire methodology. (Source: Coaches)
- [ORGANIZATION] Structured interviews and the 2-days workshop turned out to be necessary for the methodology. (Source: Coaches)

5.2.2 Negative Feedback

- [GENERAL] It is difficult to identify appropriate teaching and learning material. More emphasis on “collecting typical material” (documents, file systems, databases) would be helpful (Source: APs and Coaches).

Room for improvement: We should provide examples of good learning material for future prototypes. We rely on the experience and evaluation gained from the four application domains of the APOSDLE 2nd Prototype and we hare working towards the identification of guidelines which will help us to identify appropriate teaching and learning material for different types of application domains.

- [GENERAL] Guidelines and examples about what is an adequate learning domain/scenario for APOSDLE are missing. (Source: APs and Coaches)

  Room for improvement: In order to provide some guidelines and examples, useful to address this problem, we will study data from the evaluation of performances for the four use cases of the APOSDLE 2nd Prototype to see if it is possible to extract some useful guidelines.

5.3 Feedback on Phase 1: Knowledge Acquisition

5.3.1 Positive Feedback

- [GENERAL] Objectives of this phase are clear (Source: APs).
- [GENERAL] The candidate terms extracted automatically (with associated context) allowed the KE to validate and complete the first list of domain concepts and to provide the concept definitions (although other tools and methods than the DMT have been used to extract automatically candidate term from documents) (Source: APs).
- [GENERAL] The approach based on relevant term extraction and document clustering is appropriate (Source: Coaches).
- [GENERAL] Knowledge extracted from resources and knowledge elicited from DEs “cross-validated” each other (Source: Coaches).
- [TOOLS] Card Sorting is a useful tool for eliciting knowledge from DE. It allowed to involve people with no skills in modelling into knowledge representation. (Source: APs and Coaches).
- [TOOLS] Laddering is very useful for modelling tasks. It nearly produced the informal task model (Source: Coaches).
- [ORGANIZATION] Explanations given about this phase are sufficient for a KE (Source: APs).

5.3.2 Negative Feedback

- [GENERAL] In some application domains there are no comprehensive documents which describe an entire workflow with clear identification and labelling of single tasks and jobs.

  Room for improvement: Whether the APOSDLE system can be applied to not very structured application domains is an open issue for the APOSDLE Consortium.

- [GENERAL] Many very interesting resources are not available in digital format (Source: APs).

  Room for improvement: This is definitely true, but eliciting knowledge from non-digital resources (like paper-based document) is out of the scope of the APOSDLE Project.

- [GENERAL] File structures and document titles should be taken more into consideration (Source: APs).

  Room for improvement: Since semi-structured resources (for example, directory structures, database schemata, or content repository managers) provide useful insights of the knowledge contained in an application domain and are usually already available in many organizations, we will explore in future prototypes if, and how, semi-structured information can be used to elicit knowledge about the application domain.

- [GENERAL] Finding pre-structured documents, in order to apply a top-down-approach to build a first taxonomy of domain concepts, is not easy. However, starting bottom-up with a keyword collection extracted out of texts is too cumbersome to do it by hand (Source: APs).

  Room for improvement: Since tasks modelling seems easier than domain modelling, one possible strategy is to define first a first list (or a very initial taxonomy) of domain concepts next start with task modelling and defining the domain concepts as the knowledge required to perform a specific task.

- [GENERAL] Domain experts are always short of time. Furthermore, elicitation from experts is difficult as they do not quite understand what concept modelling is and which benefits it should bring (Source: APs).

  Room for improvement: This is a typical modelling problem, which goes beyond this methodology and the scope of the APOSDLE project. Nevertheless, the use of techniques for knowledge elicitation from experts such as laddering and card sorting can help with this problem.

- [TOOLS] The DMT tool is not adequate: the results obtained are not very useful (Source: APs).

  Room for improvement: improve the DMT.

- [TOOLS] Our experience is that card sorting tends to be less effective for people with a management role. Laddering is more effective for modelling tasks than for modelling the domain concepts. There is not a best technique to elicit knowledge from DEs. (Source: Coaches).

  Room for improvement: We will take these comments into consideration for future versions of the methodology and provide guidelines to help the coaches to select among the different techniques.

- [ORGANIZATION] A basic questionnaire (like the one used in Phase 0), adapted to the specific domain, could be helpful to elicit knowledge from DEs and could reduce the effort of the KE necessary to prepare interviews with DEs (Source: APs).

  Room for improvement: Interesting suggestion to be considered for Prototype 3.

- [ORGANIZATION] The explanations given about this phase would have been not sufficient for a novice (Source: APs).

  Room for improvement: We should improve the quality of the manual describing the steps of the methodology, and / or provide some tutorial explaining in details the phases of the methodology.

- [ORGANIZATION] No feedback has been provided by coaches/technical partners about the results of the phase to the AP's (Source: an AP).

Room for improvement: More effort will be spent for producing explicit feedback to the APs.

5.4 Feedback on Phase 2: Informal Modelling

5.4.1 Positive Feedback

- [ORGANIZATION] Very strong support by coaches (Source: APs).

5.4.2 Negative Feedback

- [GENERAL] Domain experts were not available to validate and verify the models, and they were not willing to learn about modelling. Therefore KEs and Coaches have done the validation of the models (Source: APs and Coaches).

  Room for improvement: This is a typical problem arising when performing modelling in a real-world application which goes beyond this methodology and the scope of the APOSDLE project.

- [GENERAL] The KE needs to know how the models will be or can be exploited by APOSDLE and the benefits of modelling (Source: APs and Coaches).

  Room for improvement: Some effort should be spent to provide to the KEs with a comprehensive view on why and how the models are used in APOSDLE.

- [GENERAL] It is important that also a Coach revises the models, because in the domain models or in the task models there where some mistakes the KE may not be able to detect.

  Room for improvement: We will take this comment in consideration for future prototypes.

- [GENERAL] It is hard to tell when an informal model is actually finished, and we should be clearer about this (Source: Coaches).

  Room for improvement: Examples and guidelines on how a finished informal model should look needs to be provided.

- [TOOLS] It was quite annoying that the domain concept templates were changed during modelling (Source: a MediaWiki to a Semantic MediaWiki) and the contents had to be moved by hand from the old templates to the new ones (Source: APs).

  Room for improvement: This was due to a change in tools during the modelling phase. In a future application of the methodology, we will start directly with the Semantic MediaWiki which eliminates this problem.

- [ORGANIZATION] Time consuming translation of the concepts in English language in the templates (Source: APs).

  Room for improvement: The English language translation of concepts is needed at the moment to allow the APOSDLE experts to check the models created. It will be probably removed in future prototypes.

5.4.3 Controversial Issues

5.4.3.1 Need of an informal modelling phase

The need of an informal modelling phase for APOSDLE is a controversial issue as we can see from two we received.

- [GENERAL] It is essential to start from an informal (or semi-formal) definition of models before thinking about development of OWL or YAWL models (Source: APs). This step is absolutely necessary: it is **impossible**, even if nobody believes it, to start with YAWL and Protégé directly. An alternative could be informal modelling in a graphical way (Source: Coaches).

- [GENERAL] I would prefer a mature ontology editor like Protégé, which gives a good concept overview and modelling scheme for the ontology in one interface (Source: APs).

Even if it difficult to understand if the preference of Protégé over the wiki is due to the limitations of the wiki (as used in this round of modelling - see comments in Section 5.4.3.2) or to some scepticism for the informal modelling phase, it is clear that forcing the Knowledge Engineers to describe and document their knowledge is a time consuming task, and the temptation is to start directly with producing taxonomies or process descriptions using a graphical tool, if it exists.

Therefore, while some Application Partners liked the possibility of describing the different elements (tasks and domain concepts) using Natural Language, so that the Domain Experts could check them, others found it tedious and time consuming. As a general lesson learned about this issue we can say that:

- The separation between the informal model description and its formalization has produced good formal models when the KEs liked the use of the wiki. In these case the separation between description and formalization (informal and formal steps) has produced well documented models that are easy to understand, and where the taxonomic relation (is-a), the mereonomy relation (part-of) and domain dependent relations are used in a meaningful way.
- The separation had negative effects on Application Partners who did not like the wiki and / or were already knowledgeable in formal modelling. To address this issue, we are thinking of substantially improving the wiki (see below) and also partly skip the informal modelling phase if the KE is sufficiently skilled in formal modelling. Note that the advantage of this is still debatable. In fact, an additional problem we have in building the APOSDLE Knowledge Base is that the informal modelling phase allows the integrated development of the three models (Domain, Tasks, and Learning Goals). The informal modelling phase is designed also to specify, at an informal level, an integration between these elements, for instance, via the "Knowledge Required" attribute of a task, or the classification of tasks according to task types. Defining the formal models in YAWL and OWL directly would result in skipping this early integration phase. This could generate problems as the formal models of tasks and domain could be focused on, or cover, different aspects of the application domain. For instance the task model could be very detailed on a set of tasks related to knowledge only sketched in the domain model, while the domain model could describe in detail part of the application domain not really relevant for the tasks[5]. This would make the formal integration (which was extremely effective and fast for P2) more difficult and time consuming.

5.4.3.2 The Semantic Wiki as a tool for informal modelling

The use of the Semantic Wiki as tool for informal modelling received controversial evaluations both from Application Partners and from Coaches, often about the same aspect, as we can see from the feedback below.

[5] We had to face this problem during the development of the RESCUE Knowledge Base for Prototype 1.

- [TOOLS] The Semantic Wiki helped the model revision by Domain Experts (especially for domain concepts) and the collaboration with KEs. It supported also KE's in structuring and formalizing knowledge (Source: APs).
- [TOOLS] The Semantic Wiki can be use in a collaborative manner. The APs work with the Wiki, and the coaches can support them as requested and scan the documents for errors (Source: Coaches).
- [TOOLS] The Semantic Wiki was not used very collaboratively however (Source: Coaches)
- [TOOLS] It was not obvious at first, how to get an overview (textual of graphical) of the models in the Semantic Wiki (Source: Coaches).
- [TOOLS] An explanation about how to model relations between concepts is missing (Source: APs)..
- [TOOLS] It is not quite clear how to handle the Semantic Wiki itself from a technical point of view. Should we delete obsolete or wrongly typed concepts or would there be an automatic cleaning up, for example? (Source: APs).
- [TOOLS] The Semantic Wiki functionalities and templates need to be improved (Source: APs).
- [TOOLS] The Semantic Wiki does not support the revision of filled templates in a friendly manner. This makes difficult to refine the informal models, and use an iterative process towards modelling. (Source: APs).
- [TOOLS] The Semantic Wiki is not the right tool for modelling, paper is the fastest and best understandable solution (Source: Coaches).

As an answer to all the negative comments we can say that the Semantic Wiki used for Prototype 2 was far from being ideal in many respects. This was mainly due to the fact that we moved from a normal wiki to a Semantic Wiki in the middle of the modelling phase and therefore we were not able to implement several facilities that would have simplified the handling and filling of templates. As a consequence, the workload of the KEs was heavier than necessary, in some case with negative effects on the quality of modelling. This should be addressed and improved in future versions of the methodology.

In addition to several technical problems which will be solved in future versions of the Methodology, the general lessons learned about the wiki are:

- **Positive side.** The Semantic Wiki allows the sharing of information (accessible also to DEs) and makes the extraction of formal models very easy (differently from paper descriptions)[6]; templates guide the KEs to provide complete descriptions of the model elements and also to develop the models in an integrated manner and facilitate the step of formal model integration.
- **Negative side.** The Semantic Wiki is a reasonable tool to provide a description of the single elements but is not appropriate to provide an overview (textual of graphical) of the overall models (for instance, the taxonomy of concepts or a flow of tasks). The visualization tools available for the wiki are not easily usable for our goals and this is a real limitation of using the Semantic Wiki as a modelling tool. This problem has to be solved if we want to make the use of the Semantic Wiki more effective in future versions of the methodology.

[6] For Prototype 2, two, out of four, domain ontologies were extracted almost in their final form from the SemanticWiki.

5.5 Feedback on Phase 3: Formal Modelling and Integration

5.5.1 Positive Feedback

- [GENERAL] The formal modes were created by Coaches and Technical partners. This phase was not time consuming for the APs. (Source: APs and Coaches)

- [TOOLS] Both Protégé and YAWL are nice and adequate tools for formal modelling. (Source: APs and Coaches)

- [TOOLS] Installing and using the TACT tool is very easy and the user manual contains quite clear definitions and examples of the different learning goal types. (Source: APs)

5.5.2 Negative Feedback

- [GENERAL] It is not yet clear what a “correct” or “useful” learning goal model should look like (Source: Coaches).

  Room for improvement: Starting from the evaluation of Prototype 2, we will investigate if we have enough material to compose some basic guideline to decide upon usefulness and correctness of a learning goal model.

- [GENERAL] The KEs are not able to estimate the impact of the learning goal type definition on the information retrieval and the entire APOSDLE system (Source: APs).

  Room for improvement: Some effort should be spent to provide the KEs with a comprehensive view on the impact of the models in the APOSDLE System.

- [GENERAL] Bad informal modelling in the wiki resulted in bad automatic formal model creation.

  Room for improvement: This is obviously true. To solve this, we have to improve the quality of informal models. See comments on the informal modelling phase and the use of the Semantic Wiki (Section 5.4.3.2)

- [TOOLS] There were some minor technical problems and usability issues with the TACT tool (Source: APs).

  Room for improvement: These problems have been already fixed.

- [TOOLS] TACT can be improved, for example in the visualization of the models (identification of super tasks and subtasks, relations between concepts, presentation of process instead of the alphabetical list of tasks) (Source: APs).

  Room for improvement: We will take this comment into consideration for future versions.

- [TOOLS] An automatic translation from wiki to YAWL would be helpful, but I cannot imagine a good data model for that on the wiki. (from Coaches).

Room for improvement: This is definitely an open research issue to be investigated.

5.6 Feedback on Phase 4: Validation & Revision

Validation and revision of models is going on at the time of writing. Therefore we cannot include comments on this step in this document. We will perform a careful evaluation also of this phase and use it to revise the methodology as we have done for Phases 0—3.

6 Conclusions

In this document we have described the APOSDLE Integrated Modelling Methodology. This methodology, developed within the APOSDLE project, guides the process of creation of the application domain dependent parts of the APOSDLE Knowledge Base. The APOSDLE Knowledge Base provides the basis for reasoning within the APOSDLE System.

We have provided a detailed description of the (current) first version of the Methodology, and also an overview of the lessons learned and of possible future enhancements. The feedback collected from Application Partners and coaches, who have accurately followed the current version of the Methodology to build the specific APOSDLE Knowledge Bases to be used in Prototype 2, will provide the basis of a revised version of the Integrated Modelling Methodology to be used for Prototype 3.

As we have illustrated in Section 5 the main results, coming from the definition and application of the Integrated Modelling Methodology for the production of the Application Partner Domain Models to be used in Prototype 2 are both positive and negative findings. One of the main results of the development of the Integrated Modelling Methodology is that has allowed to instantiate the general task of "modelling" inside the APSODLE context, and to identify key aspects and challenges of the modelling phase inside APOSDLE. Main challenges, which still remain to be solved, are:

- To provide guidelines for the identification of good application domains appropriate for a learn @ work tool like APOSDLE;
- To provide guidelines for the identification of the "right" granularity level of the different models;
- To ease the task of knowledge extraction by providing better automatic knowledge elicitation from digital resources. This in order to reduce the effort of Domain Experts and Knowledge Engineers;
- To improve the tools for informal modelling and to better evaluate the need of an informal modelling phase;
- To enable the Domain Experts to validate and verify the APOSDLE knowledge base, in addition to Coaches and Knowledge Engineers.

The main success of the methodology is that it has enabled the Application Partners to produce, in most of the cases, formal models of good quality, ready to be used in Prototype 2. Moreover the methodology has succeeded in helping the Knowledge Engineers to develop the different models, which compose the APSODLE knowledge base in a coherent and integrated manner. This is a crucial step for the success of the APOSDLE system. In fact a poorly integrated knowledge base, where the task model, the domain model, and the learning goal model are related by very few or too many instances of the relations depicted in Figure 1 does not contain enough (or contained too much) information to select the appropriate knowledge artefacts, starting from the context-sensitive situation of a user, thus reducing the performance of the APOSDLE system.

Bibliography

Aguado, G., Bañón, A., Bateman, J. A., Bernardos, S., Fernández, M., Gómez-Pérez, A., et al. (1998). ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation. *Proceedings of the ECAI'98 Workshop on Applications of Ontologies and Problem Solving Methods*, (pp. 1-10). Brighton, UK.

Blázquez, M., Fernández-López, M., García-Pinar, J., & Gómez-Pérez, A. (1998). Building ontologies at the knowledge level using the ontology design environment. *11th International Workshop on Knowledge Acquisition, Modelling and Management.* Banff.

Chimaera. (2000). Chimaera Ontology Environment. *www.ksl.stanford.edu/software/chimaera* .

Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). METHONTOLOGY: From Ontological Art towards Ontological Engineering. *Proc. AAAI Spring Symp. Series.* Menlo Park, California, USA: AAAI Press.

Fernández-López, M., Gómez-Pérez, A., Sierra, J. P., & Sierra, A. P. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems , 14* (1), 37-46.

Gómez-Pérez, A., & Rojas-Amaya, D. (1999). Ontological Reengineering for Reuse. *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and Management*, (pp. 139 - 156).

Gruninger, M., & Fox, M. S. (1994). The role of competency questions in enterprise engineering. *Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice.* Trondheim, Norway.

McGuinness, D. L., & Noy, N. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology.* Technical Report 2001-0880, Stanford Medical Informatics.

Ontolingua. (1997). Ontolingua System Reference Manual. *http://www-ksl-svc.stanford.edu:5915/doc/frame-editor/index.html* .

Protege. (2000). The Protege Project. *http://protege.stanford.edu* .

Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R. d., Shadbolt, N., Velde, W. V., et al. (2000). *Knowledge Engineering and Management: The CommonKADS Methodology.* MIT Press.

Sure, Y., & Studer, R. (2002). *On-To-Knowledge Methodology - Final Version.* On-To-Knowledge Deliverable 18, Institute AIFB, University of Karlsruhe.

Uschold, M., & King, M. (1995). Towards a Methodology for Building Ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95.* Montreal, Canada.

Pammer, V., Scheir, P. & Lindstaedt, S. (2007). *Two Protégé plug-ins for supporting document-based ontology engineering and ontological annotation at document-level*. 10th International Protégé Conference, 2007.

van der Aalst, W.M.P. (1998) The *Application of Petri Nets to Workflow Management*, The Journal of Circuits, Systems and Computers, Vol. **8**(1), 1998.

Godehardt, E., Doehring, M., Faatz, A., & Goertz. M. (2007). *Deploying YAWL for Workflows in Workplace-embedded Learning*, Proceedings of the 7th International Conference on Knowledge Management , Graz, 2007

Guarino, N., & Welty, C. (2004). *An overview of OntoClean*, Handbook on Ontologies, Springer 2004

Korossy. K., (1997). *Extending the theory of knowledge spaces: A competence-performance approach*. Zeitschrift für Psychologie , **205**, 1997

Noy, N.F., & McGuinness, D. (2001) *Ontology Development 101: A Guide to Creating Your First Ontology*, Knowledge Systems Laboratory, 2001

Protégé. Protégé ontology editor. *protege.stanford.edu*

Reijers, H.A. (2003). *Design and Control of Workflow Processes: Business Process Management for the Service Industry*, Springer, Berlin, 2003

Reijers, H.A. (2005): *Process Design and Redesign,* in: M. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede: Process Aware Information Systems, Wiley, 2005

Yaw. YAWL workflow language website and tool. *www.yawl-system.com*

Hewlett, D., Kalyanpur, A., Kolovski, V., & Halaschek-Wiener. C. (2005). *Effective Natural Language Paraphrasing of Ontologies on the Semantic Web*. In End User Semantic Web Interaction Workshop (ISWC 2005), 2005.

Kaljurand, K., & Fuchs, N. E. (2007) *Verbalizing OWL in Attempto Controlled English*. In Proceedings of Third International Workshop on OWL: Experiences and Directions (OWLED 2007), 2007

Jarrar, M., Keet, M., & Dongilli, P. (2006) *Multilingual verbalization of ORM conceptual models and axiomatized ontologies*. Technical report, Vrije Universiteit. 2006.

Roussey, C. (2005). *Guidelines to building ontologies: a bibliographic study*. COST C21 Technical Report.