Conceptual Framework & Architecture
Version 1
Deliverable D 2.3

Organisation name of lead contractor for this deliverable

UT – University of Twente

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

This document contains material, which is copyright of certain APOSDLE consortium parties and may not be reproduced or copied without permission. The information contained in this document is the proprietary confidential information of certain APOSDLE consortium parties and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information in this document may require a licence from the proprietor of that information.

Neither the APOSDLE consortium as a whole, nor a certain party of the APOSDLE consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using the information.

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

Imprint

Full project title: Advanced Process-Oriented Self-Directed Learning Environment
Title of work package: WP II: Self directed learning
Document title: Conceptual Framework & Architecture Version 1
Document Identifier: APOSDLE-D02.3-UT-ConceptualArchitecture_Version1
Work package leader: University of Twente
List of authors: Henny Leemkuil & Robert de Hoog (UT) (Eds.)
Stefanie Lindstaedt, Tobias Ley, Peter Scheir, Armin Ulbrich, Günter Beham, Werner Klieber (KC)
Mario Aehnelt (FHG)
Robert Lokaiczyk, Andreas Zinnen, Markus Doering, Manuel Goertz (SAP)
Luciano Serafini (FBK)
Helmuth Leitner, Barbara Kump, Viktoria Pammer (TUG)
Administrative Co-ordinator: Harald Mayer
Scientific Co-ordinator: Stefanie Lindstaedt

Copyright notice

© 2007 APOSDLE consortium
## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2007-06-15</td>
<td>First draft for inviting contributions</td>
</tr>
<tr>
<td>2.0</td>
<td>2007-10-04</td>
<td>First version with all contributions</td>
</tr>
<tr>
<td>2.1</td>
<td>2007-10-31</td>
<td>Revised version with updated contributions</td>
</tr>
<tr>
<td>2.2</td>
<td>2007-11-05</td>
<td>Final editing for submission of confidential version</td>
</tr>
<tr>
<td>3.0</td>
<td>2007-12-20</td>
<td>Final public version after incorporating comments from peer review</td>
</tr>
</tbody>
</table>
## Table of Contents

### Executive Summary ................................................................. ix

### 1 Introduction ................................................................................. 1
1.1 Purpose of this document .......................................................... 1
1.2 APOSDLE Scope and Boundaries: Focus on Prototypes 1 & 2 ........... 1
1.3 Reader Guidance ......................................................................... 4

### 2 APOSDLE Overall Approach .................................................. 5
2.1 Work-integrated learning ............................................................ 5
  2.1.1 Learning challenges ............................................................. 5
  2.1.2 APOSDLE approach: Work-integrated learning support .......... 5
2.2 Scruffy technologies to enable work-integrated learning .................. 6
  2.2.1 Technology challenges ......................................................... 6
  2.2.2 APOSDLE scruffy approach ................................................ 7
2.3 Informal learning ........................................................................ 8
2.4 Domain, task, learning goal model and their relationships .......... 8
  2.4.1 Domain model ..................................................................... 8
  2.4.2 Task model .......................................................................... 8
  2.4.3 Learning goals model: A competency-based approach for formalizing learning goals and prerequisite knowledge .............................................. 9
  2.4.4 APOSDLE knowledge base .................................................. 9
  2.4.5 Where and why do we need models in APOSDLE? ................. 9
  2.4.6 Challenges for modelling in APOSDLE .................................. 10
2.5 Context awareness and user profile services .................................. 11
  2.5.1 Context awareness and context determination ....................... 11
  2.5.2 User context representation ............................................... 11
  2.5.3 User profile services .......................................................... 12
  2.5.4 Privacy ................................................................................ 12
2.6 Work-integrated learning support ............................................. 12
  2.6.1 Supporting learning ............................................................ 12
  2.6.2 Supporting collaboration processes ...................................... 12
2.7 Knowledge artefact lifecycle .................................................... 13
  2.7.1 What is an APOSDLE knowledge artefact? ......................... 13
  2.7.2 Access to knowledge artefacts ............................................. 13
  2.7.3 Creation of knowledge artefacts ......................................... 14
  2.7.4 Retrieval of knowledge artefacts ......................................... 14

### 3 Informal Learning ........................................................................ 15
3.1 Characteristics of informal learning .......................................... 15
3.2 How can informal learning be supported? ................................ 18
  3.2.1 Just-in-time information ..................................................... 19
  3.2.2 Giving feedback and support when needed ......................... 19
  3.2.3 Stimulating reflective thought ........................................... 19
3.3 A system to support informal learning ...................................... 20
4 Domain, Task, Learning Goal Model and their Relationships .............................................21
  4.1 Domain model ..............................................................................................................21
      4.1.1 Challenges ...........................................................................................................22
      4.1.2 The APOSDLE Approach ..................................................................................23
  4.2 Task model ................................................................................................................23
      4.2.1 Challenges ...........................................................................................................23
      4.2.2 Theoretical Background ......................................................................................24
      4.2.3 APOSDLE Approach ..........................................................................................26
      4.2.4 Where else has something like this been used ......................................................27
      4.2.5 Next challenges ..................................................................................................27
  4.3 Learning goal model: a competence-based approach for formalizing learning goals and
     prerequisite knowledge .................................................................................................28
      4.3.1 Challenges for realising adaptivity in work-integrated learning with APOSDLE ....28
      4.3.2 The conceptualizing of learning goals in APOSDLE ............................................29
      4.3.3 Role of Learning Goals in APOSDLE ................................................................30
      4.3.4 The learning goal model and its theoretical background: competence-based knowledge
            space theory .............................................................................................................31
      4.3.5 Using the learning goal model in APOSDLE ..........................................................34
      4.3.6 New challenges and ideas to solve them ................................................................35
      4.3.7 Comparison of the APOSDLE Modelling Approach to Standard Industrial Competency
            Management Practice .................................................................................................35
  4.4 APOSDLE knowledge base ........................................................................................39
      4.4.1 Challenges ...........................................................................................................39
      4.4.2 Knowledge acquisition .........................................................................................39
      4.4.3 Knowledge Integration ........................................................................................40
      4.4.4 Where else has something like this been used ......................................................43
  5 Context Awareness and User Profile Services ...............................................................44
    5.1 Context awareness and context determination ..........................................................44
        5.1.1 Challenges ...........................................................................................................44
        5.1.2 Theoretical background ......................................................................................44
        5.1.3 APOSDLE approach ..........................................................................................46
        5.1.4 New challenges and ideas ....................................................................................47
    5.2 User context representation .......................................................................................47
        5.2.1 Challenge .............................................................................................................48
        5.2.2 Theoretical background and APOSDLE approach ................................................48
        5.2.3 New challenges ....................................................................................................51
    5.3 User profile services ..................................................................................................51
        5.3.1 Challenge .............................................................................................................51
        5.3.2 Theoretical background and APOSDLE approach ................................................51
        5.3.3 New challenges ....................................................................................................54
    5.4 Privacy ......................................................................................................................55
  6 Work Integrated Learning Support ................................................................................57
    6.1 Supporting learning ....................................................................................................57
        6.1.1 Challenges .............................................................................................................57
        6.1.2 Background and similar problems .......................................................................57
        6.1.3 APOSDLE approach ..........................................................................................59
        6.1.4 Remaining problems – new challenges .................................................................64
    6.2 Supporting collaboration processes ............................................................................65
6.2.1 Challenges .........................................................................................................................66
6.2.2 Contextualised Collaboration in APOSDLE .............................................................................66
6.2.3 APOSDLE collaboration framework .........................................................................................70

7 Knowledge Artefact Lifecycle ........................................................................................................71
7.1 What is an APOSDLE knowledge artefact? ..............................................................................71
7.2 Access to knowledge artefacts ....................................................................................................71
7.2.1 Challenges .............................................................................................................................71
7.2.2 Theoretical Background ........................................................................................................71
7.2.3 APOSDLE Approach ..............................................................................................................72
7.2.4 Where else has something like this been used and proven to work ........................................73
7.2.5 New challenges and ideas to solve them .................................................................................74
7.3 Creation of knowledge artefacts ..................................................................................................74
7.3.1 Challenges .............................................................................................................................74
7.3.2 APOSDLE Approach ..............................................................................................................75
7.3.3 Why did we choose this approach? .........................................................................................78
7.3.4 New challenges and ideas to solve them .................................................................................79
7.4 Retrieval of knowledge artefacts .................................................................................................80
7.4.1 Challenges .............................................................................................................................80
7.4.2 APOSDLE approach ................................................................................................................80
7.4.3 Why did we choose this approach? .........................................................................................82
7.4.4 Theoretical Background ........................................................................................................82
7.4.5 Where else has something like this been used and proven to work ........................................83
7.4.6 New challenges and ideas to solve them .................................................................................84

8 References .......................................................................................................................................86

9 Glossary ...........................................................................................................................................99
Executive Summary

The goal of the APOSDLE project is to enhance knowledge worker productivity by supporting informal learning and collaboration activities in the context of knowledge workers’ everyday work processes and within their computer based work environments. Support should be provided by means of a generic application that is not domain specific.

Achieving this goal requires conceptual as well as a technological solutions to many different challenges and problems. This deliverable describes the conceptual problems encountered and solutions proposed for the first and second prototypes of the project. Given the scope of the document it is hard to compress its message in only a few lines, so many relevant aspects are omitted from this summary.

In the most general sense, the greatest challenge is hidden in the phrase “not domain specific”. This implies that the APOSDLE system can easily be adjusted to and fit in existing work practice in a variety of organisations. More in particular, this implies that the domain of work is not known in advance, that information to be used for supporting learning at work is very heterogeneous and not specifically designed for this purpose and, finally, the information resources can reside anywhere in the document base of an organisation. Furthermore, the social network people rely on to support each other during work is not known either, nor the many different ways they interact with each other. This leads to the three major challenges listed below. For each challenge we briefly describe the kind of solution pursued in the first two prototypes of the project.

The three major challenges are:

- **Real time learning** – APOSDLE aims at supporting the knowledge worker in learning situations within her current work task. Learner support needs to be adapted to a user’s work context and her experiences, and should be short, and easy to apply. **Solution pursued** – User’s work context and current task detection by utilizing task models, automated task detection algorithms and extensive user profiling. Providing learning events that are created at the moment the worker needs them while executing a concrete work task. Embedding peer-to-peer and expert support by communication facilities that take the work and learning context into account.

- **Real computational environment** – APOSDLE aims at providing a variety of tools which are integrated seamlessly within her desktop and allow one-point access to relevant back-end systems of her organization (via some intelligent middleware). Tools need to be inconspicuous and easy to use. **Solution pursued** – Developing several computer readable models and a meta-model that allow a unified view on the work domain in the shape of a general knowledge base. Access to learning support through a single application present in a side bar.

- **Real content** – APOSDLE aims at dynamically creating learning content out of resources from the underlying organizational memory (which originally were not intended for instruction). Resources need to be compared, analyzed and retrieved based on their relationships to each other and the user context. **Solution pursued** – Creating homogeneous access to a variety of organizational resources stored in different back-end systems. Creation of learning events and retrieving of knowledge artefacts based on semantic annotations of resources. Annotations are achieved by combining manual annotation, automated methods based on machine learning principles and user feedback thereof. Associative retrieval of the organizational resources is used to find relevant material that is related to the information presented to users in learning events.

During the writing of this deliverable we were fortunate to find two members of the Advisory Board, Prof. Dr. Betty A. Collis and Prof. Dr. Jörg M. Haake, willing to review an earlier version. Thanks to their comments we believe that the document has improved significantly. Finally, it should be noted...
that later in the project the document will be updated to reflect new challenges, problems and solutions.
1 Introduction

1.1 Purpose of this document

The goal of the APOSDLE project is to enhance knowledge worker productivity by supporting informal learning and collaboration activities in the context of knowledge workers’ everyday work processes and within their computer based work environments. Support should be provided by means of a generic application that is not domain specific.

This gives rise to the following questions, which will be answered in this document:

- What are the characteristics of informal learning?
- How can informal learning be supported?
- What are meaningful collaboration activities in relation to informal learning?
- How should a generic system be organised to support informal learning?

Based on the answers to these questions the main topics of the APOSDLE conceptual architecture are identified and elaborated, namely: domain knowledge (model), task (model), learning goals (model), context awareness, categorisation and labelling of information units (in documents), supporting learning, and communication and collaboration.

This deliverable aims at providing a comprehensive overview of research challenges addressed and research approaches taken which form the basis of the software prototypes built throughout the APOSDLE project. In order to differentiate it clearly from the software architecture deliverable we named it “Conceptual Framework (& Architecture)”. Here the focus is on the underlying theories and their interactions and relationships. Specifically this deliverable is the first of two such deliverables and aims at documenting our current understanding which has influenced the development of Prototype 1 and the design of Prototype 2.

1.2 APOSDLE Scope and Boundaries: Focus on Prototypes 1 & 2

In order to understand the APOSDLE approach as laid out in this document, the reader first needs to get a better picture about what the objectives of the APOSDLE project are and specifically which kind of learning and teaching we are aiming to support.

A slightly adapted version of the APOSDLE objectives from the ‘Description of Work’ document states the following:

The APOSDLE goal is to enhance knowledge worker productivity by supporting informal learning (Chapter 3) and cooperation activities (Section 6.2) in the context (Chapter 5) of knowledge workers’ everyday work tasks (Section 4.2) and within their work environments. APOSDLE leads the way towards the seamless integration of learning, cooperation, and working at the future professional workplace.

With the term knowledge worker (here also often simply called worker) we refer to an employee of an organisation whose essential operational and value creating tasks consists in the production and distribution of knowledge (Machlup, 1962). Quite often a distinction is made between routine

1 Many writers have proposed definitions about what a “knowledge worker” is or could be (see for an overview, for example, Chapter 10 in Collins (2000)). For this report we adopt this definition of the “father” of the notion of knowledge intensive work. In particular in Task VI.5 (APOSDLE Social Solutions) and it’s deliverables, this notion will be elaborated and demarcated more precisely, based on empirical research in the participating organisations. In other words: we will move from an initial global definition of a knowledge worker to a more precisely targeted future user of APOSDLE.
knowledge work and knowledge work with a dominant creative part. Studies have revealed that truly creative activities only account for about 20% of knowledge workers’ tasks. Other types of models of knowledge-intensive work (Schreiber et al., 2000) distinguish between synthetic tasks (design, modelling, planning, scheduling, assignment) and analytic tasks (classification, assessment, diagnosis, monitoring, and prediction). Knowledge workers are predominantly controlled by overall goals and expected results instead of defined procedures. Thus, they have significant autonomy in structuring their activities (such as timing and procedures) (Pyöriä, 2003; Davenport, 2005). This means they may dynamically switch to different tasks or domains during their work. This is also reflected by dynamic changes in their user context. Knowledge workers may dynamically switch to different roles in the context of their work, for example, to that of the learner or the expert (advising other, less experienced, workers (see section 2.1 for more details).

The work environment is the set of all tools, artefacts, people, communication channels, etc. which are available to the knowledge worker at her workplace. In the context of APOSDLE our focus is on providing support within the computational work environment of the user. This focus results from the constraints we face: we are convinced that the current work task as well as the work history of the user (here simply referred to as user context) significantly influences her learning goals. Our first Workplace Learning Study (see de Hoog et al., 2006) strongly supports this belief. In order to support the knowledge worker within her learning situations we thus need to understand this user context as much as possible – and at best automatically (without having to bother the user). Automatic context determination (Chapter 5) is only possible if we assume that many important user activities take place at her computer based workplace or are represented there in some way. In turn the work-integrated learning support which we want to offer within APOSDLE, is also dependent on the computational medium.

The learning domain (or often simply called domain) refers to the domain that contains the content (knowledge) of what has to be learned or acquired during work. Within Prototypes 1 and 2 this is always exactly one domain: Requirements Engineering with Scenarios in User-Centred Environments - RESCUE (Prototype 1, Prototype 2 for CNM), Registration, Evaluation and Authorisation of Chemicals - REACH (Prototype 2 for CCI), Software Simulation for Aircraft (Prototype 2 for EADS), and Innovation Management (Prototype 2 for ISN) (see Deliverable VI.8 Application Partner Specific Models for more detail). APOSDLE is able to provide learning support only for domains formally modelled since these domain models are the basis for defining learning goals (Section 4.3) to be supported. However, the learning domain will probably only be one of many domains a user is confronted with during her work. For example, a consultant at CCI might be responsible for helping member companies adhere to the EU REACH guidelines, to help them understand Chinese import restrictions, and to comply to Basel II. In the organizational memory there will be most likely documents which deal with these three domains and many more. However, APOSDLE is focused on REACH in that the task determination facility only attempts to recognize tasks related to REACH, and the learning resources and learning events pro-actively provided, are limited to REACH knowledge artefacts (Chapter 7). Pointers are provided to people knowledgeable about certain aspects of REACH. However, in order to not totally ignore information retrieval needs of the user which might be dealing with domains other than the learning domain, APOSDLE Prototype 2 also includes a generic search mechanism which allows for keyword based retrieval of general documents and people (not leading to structured learning events).

It is unclear yet for which types of learning domains APOSDLE is specifically well suited. During modelling new aspects and challenges emerged which are addressed below in points 4 and 5. More work is needed in order to fully understand the implications. We have started to document our experiences in Deliverable VI.8 Application Partner Specific Models. In addition, we plan to provide a clearer categorization of learning domains in Deliverable VI.15 Scope of APOSDLE Target Group, Problems and Needs (M22).

2 The notion of (work) context is of course much broader than used here. The proposed limitation in this document has to do with the focus on support by means of computerized tools initially, in particular context features that can be detected automatically. In the general context of APOSDLE, Task VI.5 (APOSDLE Social Solutions) will address the more wide ranging meaning of the context concept.
Based on the above discussions the focus of APOSDLE (with specific focus on Prototypes 1 and 2) is on the following:

1. **We aim to support knowledge workers whose predominant work medium for the learning domain in focus is the computer.** This means that not only learning and selected work tasks are carried out within the computational environment, but that the majority of learning domain relevant activities takes place there. As a consequence, it must be possible to deduce the task proficiency by observing the interactions of the knowledge worker with her computer.

2. **We aim to support knowledge workers who – relevant to the learning domain in focus – are distributed over different locations.** We are well aware that many people at the workplace learn by asking their colleagues. This is a very efficient way of learning and probably will never be entirely taken over by technological support. However, in learning domains where people are not co-located, this way of learning has its drawbacks in particular because there colleagues are not always easily accessible nor are communication media always sufficiently rich to convey the needed meaning.

3. **We aim to support knowledge work and learning within domains which have a high ratio of synthetic and analytic tasks (non-routine knowledge work).** In such domains one typically does not find detailed work process models but encounters a flexible network of interrelated work tasks. Each work task here is complex and in turn requires the knowledge of a variety of specialized methods which interrelate. In addition, the way tasks are performed depend highly on the specific context in which they are performed.

4. **We are experimenting with different kinds of relationships between the learning domain and the underlying work process.** In some application cases, such as RESCUE, the learning domain and the tasks are closely related. That is, from a task one can infer the relevant domain concepts. In other application cases, such as eConsulting (ISN), the work process (customer consulting) is largely independent from learning domains (such as innovation or knowledge management). Through experimentation we have to learn which effect this has on our models and on the suitability of APOSDLE for the specific domain.

5. **We aim to support learning domains which rely to a large extent on explicit knowledge.** Much of our approach depends on making the knowledge worker aware of and providing her with the right information at the moment when she needs it and which is adapted to her level of experiences within the learning domain. This assumes that the knowledge worker possesses already many of the skills needed to apply the new knowledge within real world situations (such as having a good understanding of project management or customer interaction). Soft skills such as good presentation skills, negotiation skills, etc. will be difficult to convey with the APOSDLE approach.

6. **The focus of our work as presented in Prototypes 1 and 2 is on learning events during work processes.** We assume that a knowledge worker is busy with a specific task. During the execution of the task she realizes that she misses some knowledge or skill in order to complete it well. She then interrupts her work task and engages herself in an offered learning event (Section 6.1). During this learning event she also might have interactions with people knowledgeable about this specific problem, domain and task (Section 6.2). The interaction intended here within the first prototypes is on supporting understanding documents and learning events. This focus is empirically supported by the results of the Workplace Learning Study (see Kook en et al., 2006), which show that a large part of learning at a computational workplace is directly related to tasks a person is working on. In Prototype 3 we will also aim to support cooperation with the intention of mutual learning and knowledge construction. Due to the change in partners in WP3 this could not be achieved earlier.
1.3 Reader Guidance

In order to convey the different aspects of such a complex, interdisciplinary research activity to the reader we have chosen the following structure: Chapter 2 provides an overview of the overall APOSDLE approach, discusses the key distinctions to other approaches, and provides short overviews of the individual chapters and their interrelationships. Chapter 3 elaborates the notion of informal learning, which is at the core of APOSDLE. Chapters 4-7 address the key concepts that underlie the APOSDLE approach: domain, task and learning goal models (Chapter 4); context identification (Chapter 5); work integrated learning support Chapter 6) and the knowledge artefact lifecycle (Chapter 7). They are all structured similarly in that they address the following aspects and questions:

- Challenges
- Theoretical Background
- APOSDLE Approach
- Why did we choose this approach?
- Where else has something like this been used?
- Where has it been proven to work?
- New challenges and ideas to solve them

Chapter 8 contains all the references used in this deliverable and Chapter 9 provides a snapshot of our terminology. Since the terminology keeps evolving based on our growing understanding of the research issue itself as well as of the different research fields involved, we have set up a Wiki System which contains our terminology and which is updated regularly by the responsible partners. Please refer to https://aposdle.itc.it/glossary/index.php/Main_Page to view the current version of the glossary.
2 APOSDLE Overall Approach

APOSDLE can be characterized by the following three aspects which pose a number of research challenges:

- **Real time learning** – APOSDLE aims at supporting the knowledge worker in learning situations within her current work task. Learner support needs to be adapted to a user’s work context and her experiences, and be short, and easy to apply.

- **Real computational environment** – APOSDLE aims at providing a variety of tools which are integrated seamlessly within her desktop and allow one-point access to relevant back-end systems of her organization (via some intelligent middleware). Tools need to be inconspicuous and easy to use.

- **Real content** – APOSDLE aims at dynamically creating learning content out of resources from the underlying organizational memory (which originally were not intended for instruction). Resources need to be compared, analyzed and retrieved based on their relationships to each other and the user context.

APOSDLE addresses these three challenges from two perspectives:

1. **Learning Perspective**: Development of a paradigm for work-integrated learning
2. **Technological Perspective**: Application of scruffy technologies for learning support

The following sections first describe each of these perspectives. The chapter continues with an overview of the four major concepts in APOSDLE: informal learning, context of the knowledge worker, work-integrated learning support, knowledge artefact lifecycle.

2.1 Work-integrated learning

2.1.1 Learning challenges

In current business practice and eLearning research projects, most spending is devoted to enhancing knowledge transfer of formal training interventions. Haskell (Haskell, 1998) informs us that in 1998 US$ 70 billion were spent on formal training and Back (Back et al., 2001) states that in 2000 US$ 78 billion were spent on corporate training and continuing education. On the other hand, studies have revealed that in today’s economy only a small amount of knowledge that is actually applied to job activities (learning transfer) comes from formal training. On average people only transfer less than 30% of what is being learned in formal training to the professional workplace in a way that enhances performance. This is independent of the kind and quality of the courses taught, but mainly depends on too little consideration of work environment needs during and after formal training efforts (Robinson, 2003). 80 – 90% of what employees know of their job, they know from informal learning (Raybould, 2002). Initiatives aiming at enhancing knowledge transfer of formal training try to answer the question: “How much does the learner know after engaging in the formal training?” (knowledge transfer). Instead, as suggested by the above numbers, the question which should be asked is: “To which extent can the learner apply the newly acquired skills to her work tasks?” (learning transfer).

2.1.2 APOSDLE approach: Work-integrated learning support

Work-integrated learning happens very frequently during social interaction while knowledge workers collaborate on digital artefacts or communicate aspects of these artefacts. The role a knowledge worker embodies in social interaction, is subject to continuous change: at one point in time, a knowledge worker acts as a learner, at another point in time, the same knowledge worker herself acts as an expert (teacher) depending on her expertise with regard to the subject matter at hand (Lave &
Wenger, 1991). Hence, when we learn there is always explicitly or implicitly some teaching involved. In the case of formal training we usually encounter one teacher or trainer who conveys the content to be learned. But in other situations – such as reviewing code – this teaching role is not so obvious to the expert herself. In the following we will refer to such episodes as *work-integrated teaching*.

The key distinction of the APOSDLE approach, as compared to more traditional (e)Learning approaches, is that APOSDLE will provide integrated support for the three roles a knowledge worker fills at the professional workplace: the role of learner, the role of expert, and the role of worker.

**Learner Support.** APOSDLE provides learners with support for self-directed exploration and application of knowledge. This is done within their work environment such that learning takes place within the learner’s current work context. APOSDLE provides learners with guidance through the available knowledge by applying novel learning strategies. Content from knowledge sources are presented to learners, even if the content provided has originally not been intended for learning.

**Expert Support.** APOSDLE acknowledges that most effective learning transfer happens during communication, collaboration and social interaction. APOSDLE lowers the hurdles for knowledge workers to informally convey knowledge via their computational environment in that it captures the context of the creation, evolution and usage of artefacts. APOSDLE enriches artefacts with context information and thereby allows artefacts to be turned into true learning artefacts (contextualized collaboration).

**Worker Support.** APOSDLE tightly incorporates learning and teaching episodes into the work processes in that it takes care of several aspects of workers' work contexts, such as a worker’s knowledge state, work situation, and application domain. Workers are provided with context sensitive knowledge, thus raising their own awareness of learning situations, content, and people that may be useful for learning. APOSDLE enables workers to access content from several diverse knowledge sources without having to change the work and learning environment.

To work, learn and teach efficiently and effectively, a knowledge worker must be provided with optimal guidance to manage the large variety of knowledge artefacts available in the corporate information infrastructure. Therefore, the seamless integration of the underlying information spaces into an integrated semantic knowledge structure is of paramount importance. APOSDLE will therefore create such an infrastructure (referred to as the APOSDLE platform) to support the integration of the three roles.

2.2 *Scruffy* technologies to enable work-integrated learning

2.2.1 Technology challenges

Typically eLearning systems are a wonder of carefully designed content, fine-granular models, interdependencies and hand crafted metadata: The learning domain is broken down into meaningful learning units or modules which encompass concepts, facts and processes. They entail fine granular learning information, exercises, tests, etc. Each of these units is carefully designed using a multitude of different media appropriate for the learning type and learning purpose the unit is serving. A dependency structure identifies prerequisites and post-conditions. Based on the units, learning paths (courses) can be created by instructional designers taking into account the target group as well as preferred didactical aspects. In order to allow for improved personalization, a multitude of metadata is attached to the units. In addition, eLearning systems provide detailed user models which allow for the

---

3 The term scruffy stems from Artificial Intelligence where a distinction is made between neats and scruffies. Neats consider that solutions should be elegant, clear and provably correct. Scruffies believe that intelligence is too complicated (or computationally intractable) to be solved with the sorts of homogeneous system such neat requirements usually mandate. The distinction was originally made by Roger Schank in the mid 70s to characterize the difference between his work on natural language processing from the work of John McCarthy, Alan Newell and others whose work was based on logic.
representation of different learning levels in the different areas, learning preferences, etc. Tutors and teachers are represented in order to allow students access to expert help. Not to speak of class and lecture management, simulation and games, etc.

In short, one is faced with a thoroughly designed network of interrelated pieces which need to be artfully concerted to deliver a meaningful learning experience to the user. Reflecting on these properties, one can easily understand why eLearning content is expensive to create, requires lots of (metadata) standardization, and also requires a lot of organizational structure.

In contrast, new learning approaches such as work-integrated learning (see Lindstaedt & Mayer, 2006, for possible scenarios) and organizational learning put one requirement in the centre of attention: Flexibility. Being closer to the application of knowledge (rather than to the internalization of knowledge) such approaches critically rely on providing always the newest available content in ever changing learning situations. While in traditional course-oriented eLearning one could still manage the large amount of design work (also because the learning domains stayed rather stable), this is not the case any more in these new settings. Here we have to strive for the best possible available learning information instead of striving for the best designed eLearning content.

Thus, in such situations it is simply impossible to create and maintain such a carefully crafted network of interdependent learning pieces and structures. Instead, we have to move towards embracing approaches which enable us to best deal with change – while at the same time accepting their side effects such as a lower level of accuracy, likelihood of errors and not always optimal instructional design.

With APOSDLE, we present possibilities of moving away from the pure and neat approaches of instructional design (based on hand crafted verified formal models) to the application of scruffy technologies (hybrid approaches which also take context into account) to enable work-integrated learning. The “intelligence” within such systems may be “seen as a form of search and as such not perfectly solvable in a reasonable amount of time” (Gigerenzer & Todd, 1999).

### 2.2.2 APOSDLE scruffy approach

The foundation for the APOSDLE approach is to not rely on specifically created (e)Learning content, but to reuse existing (organizational) content which was not necessarily created with teaching in mind. We tap into all the resources of an organizational memory which might encompass project reports, studies, notes, intermediate results, plans, graphics, etc. as well as dedicated learning resources (if available) such as course descriptions, handouts and (e)Learning modules. The challenge we are addressing is: How can we make this confusing mix of information accessible to the knowledge worker in a way that she can advance her competencies with it?

A frequently travelled path (also within eLearning systems) is the creation of fine-grained semantic models which allow for the categorization and retrieval of such resources. But as we discussed above, the creation of such models, their maintenance and the annotation of resources with their concepts prove prohibitive in a dynamic environment. Thus, the APOSDLE approach is a hybrid one: complementing coarse grained semantic models (maintained as much as possible automatically, see below) with the power of diverse associative methodologies, improved over time through usage data and user feedback (collective intelligence).

Here the models play two roles: serving as initial retrieval triggers and providing the basis for simple inferences and heuristics to interpret user interactions. A disadvantage of this approach is that “statements” made by the system such as “this resource helps you to understand the concept of use case modelling” or “this person has expertise in use case writing” rely on empirical observations with no claim to accuracy. However, users have become increasingly accustomed to this concept through their usage of (internet) search engines. Also, obsolete models do not provide any added value and additionally are in danger of providing a false sense of security.

The APOSDLE approach is to apply a battery of advanced scruffy technologies to bridge the gap between coarse grained models and fine grained learning needs. The ultimate goal of this research is
to minimize or at best fully eliminate the need for formal models. This will also significantly reduce the amount of human effort needed to create eLearning systems.

2.3 Informal learning

In Chapter 3 we will discuss the informal learning theories which serve as points of departure for the development of the work-integrated learning paradigm. By this, the following chapters are grounded into learning theories.

Specific focus is given to Knowles’ Self-Directed Learning (SDL) definition. SDL characteristics are explored and it is discussed how the different possible learning routes can be supported through instructional design. Specifically three instructional design paradigms are taken into consideration: just-in-time information, giving feedback and support when needed, and stimulating reflective thought.

A combination of these three instructional design paradigms leads finally to the definition of high level requirements for a SDL support system such as APOSDLE.

For Prototypes 1 and 2 the extent to which communication and collaboration between users is taken into account is limited to providing advice additionally to the written resources delivered. This is due to the partner change in WP3. For Prototype 3 the role of collaboration, mutual learning and collective knowledge construction will be strengthened.

2.4 Domain, task, learning goal model and their relationships

Chapter 4 explains the three model structures and their relationships which provide the basis for reasoning within APOSDLE:

- Domain model – provides a representation of the learning domain in OWL format (OWL is a Web Ontology Language)
- Task model – provides a representation of the work tasks to be supported in YAWL (Yet Another Workflow Language) format
- Learning goal model – provides a mapping between domain concepts, tasks and general learning goal types

2.4.1 Domain model

The objective of the domain model described in Section 4.1, is to provide a semantic/logic description of the learning/work domain of an APOSDLE deployment environment. The domain is described in terms of concepts, relations, and objects that are relevant for this domain. Technically speaking the domain model is an ontology that defines a set of meaningful terms which are relevant for the domain and, which are used to classify and retrieve knowledge artefacts

2.4.2 Task model

The objective of the task model described in Section 4.2 is to provide a formal description of the tasks the knowledge worker can perform in a particular domain. The section starts with an overview of the business process modelling history. It leads the reader through different process presentations to the use of YAWL to define workflows. Reasons are provided why YAWL was chosen as a modelling language for tasks and their sequencing within processes for APOSDLE.
2.4.3 Learning goals model: A competency-based approach for formalizing learning goals and prerequisite knowledge

Section 4.3 explains the mapping between tasks and learning goals in order to realize an adaptive system. A learning goal describes knowledge and skills needed to perform a task. It is defined as a discrete element of a cognitive activity (learning goal type) connected with a domain concept. The formalisms employed are based on competence-based knowledge space theory and they provide several advantages for APOSDEL. One important advantage is that it allows the computation of learning goals through a learning need analysis by comparing knowledge needed to execute a task and the knowledge state of the user. Another one is the possibility to infer a user's learning history by examining the work task she has engaged with in the past (task-based learning history). A final advantage of utilizing knowledge space theory within APOSDEL is that the mappings afford the computation of prerequisite relationships between learning goals. This allows APOSDEL to identify learning goals which should be mastered by the user on the way to reaching a higher level learning goal.

2.4.4 APOSDEL knowledge base

This section described the APOSDEL Knowledge Base (AKB) which contains the three different models briefly discussed above, and the meta-model schema interrelating them. Within this AKB each model is stored in its original format (OWL, YAWL) and also within the meta-model schema. The meta-model schema is an ontology represented in OWL. The advantage of keeping the models in both representations is that within the original representation (for example, YAWL for tasks) reasoning about processes is possible while within the OWL representation reasoning about the overall model is supported.

2.4.5 Where and why do we need models in APOSDEL?

Figure 2-1 below illustrates the information flow through APOSDEL, with two alternative paths: the “Learning Tool Path” uses the available information to arrange available resources in a way that is adapted to support learning for the knowledge worker. The “Related Resources Path” uses the scruffy associative retrieval to find resources that are possibly related to the knowledge worker’s current situation (in task, domain and learning need).

![Figure 2-1: Information flow in APOSDEL](image-url)
A knowledge worker is working in some task (1). In order for APOSDLE to start delivering useful information and support for the user, the task must be detected (2). This can be done either manually or automatically. In both cases, the task must have been modelled in the task model.

- If the knowledge worker manually selects the task(s) he is in, the tasks must be described verbally in an intelligible way. Also, the task must be at a granularity that makes sense to the user, that is, neither very specific nor highly abstract.
- If the task is detected automatically, the task again must be at a fitting granularity level, but this time it must "make sense to the detection algorithm". This is given, for example if the task can be clearly expressed by a set of detectable human-computer-interactions.

To each task a set of learning goals are related (3). These learning goals are modelled in the learning goal model. Obviously, it is necessary, that to each task in the task model at least one learning goal is related, otherwise, the information flow within APOSDLE “breaks” here.

The set of learning goals required by the selected tasks are ranked (4) according to various criteria. One criterion is prerequisite relations between learning goals (see Section 2.4.3). In order to derive prerequisite relations between different learning goals, it is necessary that some learning goals are associated with more tasks. For ranking of learning goals it is therefore advantageous if there are many tasks but few learning goals.

The knowledge worker manually selects one learning goal (5). In order to be able to do so, the learning goal must be intelligible to the user. The learning goal consists of a learning goal type and a domain model element. The domain element is modelled in the domain model. This means that the learning goal types must be well described, the domain model element must be well described verbally and the semantic of a learning goal must be clear (that is, what does the combination of such a learning goal type and a domain model element mean).

Based on the learning goal type of the selected learning goal, a learning template (see Section 6.1) is chosen (6a). Depending on the learning template several queries consisting of material uses and domain model elements (each query consists of one material use and one domain model element) are sent to the APOSDLE system (7a). The APOSDLE system retrieves exact matches for the query. The annotations of the knowledge artefacts therefore must consist of domain model elements that occur in the learning goals and material use types associated with the selected learning template(s).

The APOSDLE system also offers resources that are related to the selected learning goal (6b). The associative retrieval uses the domain model element part of the learning goal to search for resources based on an associative network that connects similar domain model elements and similar documents (see Section 7.4.2). The associative network assumes correct semantics from the domain model (especially with respect to the hierarchical relations). The associative network uses different similarity measures depending on the domain model’s characteristics. In general, a more complex structure of the domain model allows the application of more different similarity measures, which can improve the retrieval of relevant knowledge artefacts.

2.4.6 Challenges for modelling in APOSDLE

The challenges related to the role of models within APOSDLE mainly refer to the problem of conceptually integrating the three separate models each having their own representation, as well as integrating the “semantics of APOSDLE”, that is the way APOSDLE interprets the models’ meanings given the intention of the models.

- Tasks, domain model elements and learning goals need to be modeled at some intermediate level of granularity. How this level of granularity can be described more precisely, is a question for research.
- At various points in the information flow described in Figure 2-1, different parts of APOSDLE have potentially conflicting requirements on the models. For example, whereas for learning goal ranking, many tasks and few domain model elements would
be advantageous, knowledge artefact retrieval only makes use of the domain model elements, and so fewer domain model elements lead to potentially less differentiated view on the knowledge artefacts.

2.5 Context awareness and user profile services

User context determination plays a crucial role in the overall APOSDLE approach. In order to be able to provide the user with information, learning material and links to people relevant to her task at hand, the system needs to identify the work task reliably. The identified task then updates the user profile (Section 5.3) and causes a number of activities to be started pro-actively: re-computation of a user’s learning goal and learning goal histories (Section 4.3), search of knowledge artefacts relevant to the learning goal (Section 7.4), search of people relevant to the learning goal (Chapter 6.2), and the dynamic creation of learning events (Section 6.1). The results of these searches are displayed in a resource list and a people list. Chapter 5 deals with the task detection process and the user profile services.

2.5.1 Context awareness and context determination

Section 5.1 explains and stresses the importance of context awareness for eLearning systems in general and APOSDLE in particular. We start with providing the reader the vision of context aware systems, its origin, and current developments in this field. Specifically a number of context aware systems are discussed. The APOSDLE approach to context awareness in based on a common architecture by Baldauf et al. (2007) which includes the use of agents in three layers separating the detection of context, planning and action based on context.

The section also details the challenges of user context determination. The goal is to identify a user’s current work task based on the user’s interactions with the system (key strokes, mouse movements, applications used, etc.) and the metadata and content of the resources accessed (mail messages, documents, links to people, etc.).

The APOSDLE approach to context determination involves two phases: a training phase and a run-time phase. During the training phase task executions by a number of different users are captured and labelled. These captured execution logs are then utilized to train a classifier to distinguish between the different tasks. During run-time the context determinator continuously monitors the interactions of the user and tries to automatically classify the execution logs. If an execution log is classified with a confidence value above a certain threshold the task is recognized.

While already taken into account during the current design, feedback mechanisms are still in their infancy. These will be further explored within the second version of this deliverable and implemented in Prototype 3.

2.5.2 User context representation

APOSDLE stores user related context information in digital user profiles. These profiles are used for maintaining the user's usage history and current context with respect to their personal work-, learning- and collaboration-related experiences. The APOSDLE approach differentiates between four forms of user related data: user data, usage data, inferred data, and environment data. This layering of user profile information allows us to clearly separate factual information and assumed information about the user. The outermost layer (environment data) is actually not directly related to the user profile and is as such not stored within the user profile. Much of it is stored instead in the APOSDLE Knowledge Base (see Section 4.4). Nevertheless, environment data have a significant impact on the user profile (specifically on the inferred data) and therefore need to be considered and explained in this section.

APOSDLE uses data stored in the user profiles for adapting its support to the users’ needs and requirements. Based on the user profile data, recommendations are computed aiming at supporting
the users’ learning goal attainment, the preparation of the retrieval of resources (Chapter 7.4) and acts of collaboration (Chapter 6.2).

### 2.5.3 User profile services

The component responsible for operations upon user profiles is the User Profile Service (UPS). The UPS’ functionality is made accessible to other parts of the APOSDLE system via a set of services. Examples are services for logging users’ activities and services computing queries for resources, which optimally fulfil a user’s current learning need. The UPS’ contextualized services will be described.

### 2.5.4 Privacy

Storage and manipulation of user related information of course raises critical issues concerning privacy and ethics in general. These issues are tackled by giving users large control over their own personal data.

### 2.6 Work-integrated learning support

Chapter 6 describes the concrete support mechanisms developed in Prototypes 1 and 2 for the three roles the knowledge worker assumes during work: worker, learner and expert.

#### 2.6.1 Supporting learning

Based on the chosen SDL learning paradigm, the challenges for the automatic domain-independent creation of instructional material from organizational resources (typically created not with instructional intentions in mind) are presented and approaches to similar problems are discussed.

The APOSDLE approach to these challenges involves the design of generic (domain independent) learning templates for different learning goal types which during run-time are automatically filled with relevant material from the organizational memory. The chosen learning goal types are adapted from Anderson & Krathwohl’s (2001) learning (sub) processes and determine the type of material which should be included in the final learning event. This requires that knowledge artefacts should be annotated with a material use type. These material use types include examples, definitions, procedures etc.

In Prototype 1 a learning template consists of a header, a slot for retrieved content, and an engagement activity. In Prototype 2 a learning template also contains possibilities for the user to give feedback on the quality of the content and on the overall learning event. In addition, in Prototype 2 the learner can indicate if she thinks that she has acquired the new knowledge or skill.

Section 6.1 then continues by describing the exact mechanisms, with which the learning tool chooses the relevant learning templates, fills them with knowledge artefacts (domain knowledge) and presents them to the user via the resource list (Prototype 2) or sidebar (Prototype 1).

Section 6.1.4 about remaining problems highlights that communication with experts and other learners needs to be designed more detailed into the learning templates. This will be a specific challenge for Prototype 3.

#### 2.6.2 Supporting collaboration processes

Collaboration support within APOSDLE (see Section 6.2) focuses specifically on communication and collaboration support in learning situations. That is, it is not intended to support the entire palette of possible user-to-user interactions, but aims to investigate which interactions are especially important during a learner-expert exchange.
This section uses an activity model to illustrate the different possible collaboration/communication activities within an APOSDLE community, not all of which are supported in Prototypes 1 and 2. Depending on the type of interaction desired, relevant channel selection needs to take place. The APOSDLE approach is to first help the user choose the right person to talk to, and then to provide the user with a choice of different communication channels depending on several aspects like availability of the person to be contacted, preferences of the person to be contacted, affordances of the learning situation to be supported, etc.

Another important APOSDLE approach to communication is the contextualization of communication channels. Instead of simply offering the channels, APOSDLE aims to initialize them by conveying the context of the learner to the person being contacted (expert). This includes providing information about the current work task being executed by the learner, the relevant learning events visited by the learner, relevant documents accessed and open, etc.

In order to enable other users to also learn from an interaction between a learner and an expert, it should be possible to make (edited) transcripts of the communication available. This topic has been moved to Prototype 3 since, due to the partner change, the details of the collaboration and cooperation support in APOSDLE, were not available at the time of writing of this Deliverable.

2.7 Knowledge artefact lifecycle

Chapter 7 describes the lifecycle of knowledge artefacts: first a document enters the APOSDLE system as a repository object (for example, during system instantiation, during nightly updates, after communication events). Next, the entire document, or parts thereof, are turned into knowledge artefacts by manually or automatically attaching two types of metadata: domain concept(s) present and material use of the knowledge artefact. Finally, the knowledge artefacts are related to each other and retrieved via an associative network. First ideas are included on how user feedback mechanisms during the different steps can be utilized. The individual steps are described in more detail below.

2.7.1 What is an APOSDLE knowledge artefact?

Section 7.1 gives a formal definition of what we in APOSDLE refer to as a “knowledge artefact”. Informally speaking, a knowledge artefact is a document or part thereof together with two types of metadata: the learning domain concept addressed/described within the document (piece) and the material use type of the document (piece). Optionally a knowledge artefact can also have a free text comment associated with it.

2.7.2 Access to knowledge artefacts

Section 7.2 addresses the challenges APOSDLE faces when providing access to a number of different back-end systems of the organization. The two main challenges are unified access and privacy aspects.

Based on technologies like GRID computing, APOSDLE allows homogeneous access to all documents stored within the attached back-end systems via a Data Object Repository. Within this repository each document is represented by a repository object. This repository object can be understood as a document representation enriched with metadata. They are stored in a database within the APOSDLE platform for further access by other APOSDLE modules. The repository manager deals with the repositories within one APOSDLE system.

Concerning privacy, underlying repository objects is an access mechanism which takes the permissions of the corresponding document within the back-end system into account. Mapping techniques are applied to resolve the access permissions.
2.7.3 Creation of knowledge artefacts

One essential step in the APOSLDE knowledge artefact lifecycle is how knowledge artefacts are created. One approach to creating knowledge artefacts is to do this in a collaborative fashion, where users create knowledge artefacts by annotating documents with a domain concept and/or a material use as described below. Due to the huge amount of knowledge items targeted by APOSLDE, a single user or a small user group will not be capable to annotate all knowledge artefacts. Therefore, tools must support a collaborative annotation so that users can share their knowledge about knowledge artefacts with each other.

However, manual annotation is a labour intensive task and we do not necessarily expect manual annotations to provide training data covering all necessary concepts for automatic classification. To overcome this natural limitation and to bootstrap an annotation free repository, APOSLDE takes use of linguistic and machine learning approaches to automatically assign domain concepts and material uses to knowledge artefacts. These issues are dealt with in Section 7.3.

2.7.4 Retrieval of knowledge artefacts

In order to provide powerful, intelligent retrieval mechanisms to the work integrated learning support tools (Chapter 6), the APOSLDE approach includes an associative network. This associative network implements heterogeneous retrieval mechanisms: semantic retrieval (based on learning domain concepts) is seamlessly integrated with a variety of similarity-based retrieval mechanisms. This has the advantage of on the one hand providing the learning tool with exact matched materials and on the other hand also providing more in-exact similarity-based results for work and learning support. In addition, the fact that associative networks can “learn” based on changing the edge weights, is used by APOSLDE to incorporate implicit as well as explicit user feedback.

The challenges identified in this chapter stress this particular use of the associative network in APOSLDE. An additional challenge is the sparse annotation of knowledge artefacts within the repository, since annotating documents or parts thereof with metadata is an effort intensive process. In APOSLDE we have taken two approaches to solve it. On the one hand we employ automatic annotation mechanisms as much as possible. On the other hand, we acknowledge that not all knowledge artefacts will have high quality metadata attached. By utilizing text-based and multi-media based similarity metrics, APOSLDE is able to identify knowledge artefacts which are similar to a set of previously retrieved knowledge artefacts (retrieved based on domain concept(s)) and to identify domain concepts which are similar to the original one.
3 Informal Learning

As informal learning is at the core of the APOSDLE project, a brief overview of this topic is presented in this Chapter.

3.1 Characteristics of informal learning

In 1968, Malcolm Knowles proposed “a new label and a new technology” of adult learning to distinguish it from pre-adult schooling (p. 351). The concept of andragogy, which he defined as “the art and science of helping adults learn,” was contrasted with pedagogy, the art and science of helping children learn (Knowles, 1980, p. 43). The five assumptions underlying andragogy describe the adult learner as someone who (1) has an independent self-concept and who can direct his or her own learning, (2) has accumulated a reservoir of life experiences that is a rich resource for learning, (3) has learning needs closely related to changing social roles, (4) is problem-centred and interested in immediate application of knowledge, and (5) is motivated to learn by internal rather than external factors. The concept of andragogy refers to formal as well as informal learning. Formal learning is planned learning that derives from activities within a structured learning setting. Eraut (2000) distinguishes the following five characteristics of formal learning:

- A prescribed learning framework.
- An organised learning event or package.
- The presence of a designated teacher or trainer.
- The award of a qualification or credit.
- The external specification of outcomes.

According to him, informal learning lacks these characteristics, which makes it quite distinct from more “orderly” formal learning. Dale and Bell (1999) define informal learning as: “Learning which takes place in the work context, relates to an individual’s performance of their job and/or their employability, and which is not formally organized into a programme or curriculum by the employer. It may be recognized by the different parties involved, and may or may not be specifically encouraged”. Marsick and Volpe (1999) have a slightly different view and give the following characteristics of informal learning (the third and fourth aspect are not in line with the definition given above):

- It is integrated with daily routines.
- It is triggered by an internal or external jolt.
- It is not highly conscious.
- It is haphazard and influenced by chance.
- It is an inductive process of reflection and action.
- It is linked to learning of others.

The ideas of informal learning often are referred to with other terms like self-directed learning, experiential learning, non-formal learning, implicit learning, and workplace-focused learning. In the APOSDLE project we use the term Self-Directed Learning (SDL). This concept is already used for a very long time. From literature we can infer that SDL was already used in classical antiquity (Brockett & Hiemstra, 1991). Because it is such a commonly used concept, several definitions of SDL do exist. Correspondingly, there is no consensus about the exact meaning of SDL.

Strictly speaking, any form of education has some form of self-direction in it. However, no act of learning is fully self-directed. Within APOSDLE we follow the most commonly used definition of
Knowles. Knowles (1975) regards SDL as a process in which individuals take the initiative in designing learning experiences, diagnosing learning needs, locating resources, and evaluate learning. As stated above the idea behind Knowles’ theory is that adults have fundamental different educational characteristics than children. Instruction for adults should be tailored to these characteristics and associated learning needs. The learning process in SDL consists of five steps:

1. Diagnosing learning needs
2. Formulating learning goals
3. Identifying human and material resources for learning.
4. Choosing and implementing an appropriate learning strategy
5. Evaluation of the learning outcome.

A learning need arises when knowledge workers are in doubt about the knowledge that is needed to give meaning to an event. They acknowledge that there is a gap between the knowledge needed and knowledge they possess. These feelings of uncertainty are converted into concrete subjects or questions that lead to searching for information or knowledge (Choo, 1998). From a learning need, more concrete learning goals can be derived. As Merriam and Cafferella (1991) comment, this means of conceptualizing the way we learn on our own is very similar to much of the literature on planning and carrying out instruction for adults in formal institutional settings. It is represented as a linear process. However, adults do not necessarily follow such a well defined set of steps - but are far more liable to chance and circumstances.

Spear and Mocker (1984) found that “self-directed learners, rather than pre-planning their learning projects, tend to select a course from limited alternatives which happen to occur in their environment and which tend to structure their learning projects”, or, in other words, an event or phenomenon triggers learning. Self-directed learning becomes possible, when context and individual factors coalesce to form the stimulus and the opportunity for reflection and exploration. When this happens a potential learning situation arises.

Jarvis (1987, 1995) set out to show that there are a number of responses to a potential learning situation. He used Kolb’s model of experiential learning with a number of different adult groups and asked them to explore it based on their own experience of learning. He was then able to develop a model (see Figure 3-1) with different routes. Some of these represent non-learning, some non-reflective learning, and some reflective learning.
Jarvis (1987) set out nine routes of responses to a potential learning situation. The first three routes are the ones in which no learning takes place:

1. Presumption (boxes 1-4): people react through mechanical responses or a presumption that what has previously worked will work again
2. Non-consideration (boxes 1-4): when a person does not respond to a potential learning situation.
3. Rejection (boxes 1-3 to 7 to 4): when a person chooses to reject the opportunity to learn.

Level Two: Non-reflective learning:
4. Pre-conscious (boxes 1-3 to 6 to 9): when having experiences in daily living that are not really thought about
5. Practice (boxes 1-3 to 5 to 8 to 6 to 9): when a person practices a new skill until it is learned like training for a particular physical skill or the acquisition of a language.
6. Memorization (boxes 1-3 to 8 to 6 and then to 9): when a person evaluates presented information and chooses to memorise it so it can be reproduced at a later time.

Level Three: Reflective learning:
7. Contemplation (boxes 1-3 to 7 to 8 to 6 to 9): when a person thinks about what is being learned.
8. Reflective practice (boxes 1-3 (to 5) to 7 to 5 to 6 to 9): when there is reflection on and in action such as in problem solving.
9. Experiential learning (boxes 1-3 to 7 to 5 to 7 to 8 to 6 to 9): when there is actual experimenting on one’s environment, which might be the way to learn pragmatic knowledge.

Although some remarks can be made about the routes Jarvis distinguishes (for instance: why is there no route with a direct connection between box 8 and box 4), Jarvis model illustrates that not every potential learning situation leads to learning and that a distinction needs to be made between reflective
and non-reflective learning. When non reflective learning occurs, mostly the persons are not aware that they have learned something and knowledge is often implicit and difficult to verbalise. This means that they can apply the knowledge in situations that resemble the situation in which they have learned it, but have difficulties in explaining what they did (and why they did it) to other people. Furthermore, they often have difficulties in transferring that knowledge to other contexts. For example: when typists were given caps for typewriter keys, they could not arrange them in the proper configuration, yet they all could type rapidly and accurately (Norman, 1988).

The distinction made by Jarvis resembles the one made by Norman (1993). Norman describes two modes of cognition: an experiential mode and a reflective mode. “These two modes do not capture all of thought, nor are they completely independent” (p.16). The experiential mode is one of perceptual processing, it is a pattern or event driven activity. It requires some thought but the information processing is data driven and reactive. According to Norman, this mode leads to an accumulation of facts, it reactivates information that is already present in the memory system and it leads to a tuning and shaping of knowledge structures already available.

“The reflective mode is that of comparison and contrast, of thought, of decision making” (p. 16). It is slow and laborious. “Reflective thought requires the ability to store temporary results, to make inferences from stored knowledge and to follow chains of reasoning backward and forward, sometimes backtracking when a promising line of thought proves to be unfruitful……The use of external aids facilitates the reflective process by acting as external memory storage, allowing deeper chains of reasoning over longer periods of time than possible without the aids” (p. 25). Effective reflection requires some structure and organization and is greatly aided by systematic procedures and methods and the aid of other people.

3.2 How can informal learning be supported?

Because of its nature, it is difficult to support informal learning. Traditional instructional design models are not well suited because it is difficult to foresee what kind of learning need will arise, what type of knowledge is involved and what the level of prerequisite knowledge is that the learner already has mastered. To determine how we can support informal learning, one has to have an idea about what knowledge workers do when a learning need arises. Cross and Parker (2004) found that knowledge workers typically spend a third of their time looking for information and helping their colleagues do the same. According to them, a knowledge worker is five times more likely to turn to another person rather than an impersonal source such as a database or knowledge management system, and only one in five knowledge workers consistently finds the information needed to do their jobs.

According to Dalkir (2005), other people are the preferred source of information for a number of reasons. One is that it is often faster. Another is that when one turns to another person, one not only ends up with the information but one also learns where it is to be found, how to reformulate the initial question, whether one was at the right track, and where one strayed. Last but not least, the information is coming from a known, trusted and credible source.

The data presented above (and to a large extent confirmed in the 1st Workplace Learning Study conducted in the project, see de Hoog et al., 2006) indicate that support for informal learning by knowledge workers should focus on helping to find the right knowledge and the right colleagues that could be of help when a potential learning situation arises. Furthermore the model of Jarvis stresses the idea of reflection as an important aspect of learning.

This relates to three instructional design paradigms:

- Just-in-time information presentation
- Giving feedback and support when needed
- Stimulating reflective thought

These will be elaborated below.
3.2.1 Just-in-time information

It is known from the literature that the presentation of relevant knowledge is most effective when the information is offered Just-in-Time (see de Jong & van Joolingen, 1998; Kester, Kirschner, van Merrienboer, & Baumer, 2001). The rationale is that the necessary information must be active in working memory when (a part of a complex) task is practised. In order to reach this goal, the information can best be presented when it is needed – or at least be directly available during practice. Organisation of information in small units is assumed to be appropriate because only the presentation of relatively small units can prevent processing overload. Information units may comprise several elements, like:

- The procedural steps or rules that describe the correct performance of the skill under consideration (how to do it)
- The knowledge that is prerequisite to performing the step or rule.
- Examples that illustrate the procedure/concepts.
- Feedback on performance of the steps.

A step-by-step approach is only possible when the instructional designer has control over the task that is being performed. When this is not possible, for instance when the learner has a large degree of freedom to perform the task just-time information presentation may be applied by implementing learning aids (like help systems, (minimal) manuals, checklists) that make the information that is needed easily accessible at the moment it is needed.

A specific type of help system that focuses on workplace learning is an Electronic Performance Support System (EPSS). According to Gery (1991) an EPSS is "an integrated electronic environment that is available to and easily accessible by each employee and is structured to provide immediate, individualized on-line access to the full range of information, software, guidance, advice and assistance, data, images, tools, and assessment and monitoring systems to permit job performance with minimal support and intervention by others". Looking at this definition one can conclude that this is also what APOSDLE is aiming at. The difference between an EPSS and the APOSDLE system is that an EPSS is developed to support a specific cluster of task domains, while the APOSDLE system is generic in nature, which creates new problems and challenges. Furthermore, the goal of APOSDLE is not only to support performance, but also to support learning.

3.2.2 Giving feedback and support when needed

The idea of feedback and support when needed refers to traditional workplace learning in which an apprentice learned a trade such as tailoring or woodworking by working under a master teacher. By observing the master and by listening to the master explaining exactly what he is doing and thinking as he models the skill, the apprentice can identify relevant behaviours and develop a conceptual model of the processes involved. The apprentice then attempts to imitate those behaviours with the master observing and providing coaching by means of feedback. At the end of the 20th century the importance of this approach was refreshed in the cognitive apprenticeship approach to teach complex cognitive skills (Collins, Brown, & Newman, 1987) which stressed the processes of coaching, articulation, exploration and reflection.

Eraut and Bierma (2004) recently acknowledged the role of support and feedback in workplace learning. He indicates that his research showed that in many contexts informal support from whoever was available was more important for workplace learning than formally designated helpers. Furthermore, that giving and receiving feedback often was vital. He found that learners needed both short-term task-specific feedback and longer-term more strategic feedback on general progress.

3.2.3 Stimulating reflective thought

Supplying information, feedback and support when needed in many cases will be sufficient to solve the problem a knowledge worker was facing while performing a specific task. However, this does not
necessarily mean that the knowledge worker has learned something in the sense that he has changed existing knowledge structures or has acquired new structures that enable him to perform the task in the future without support. For this latter to happen, active reasoning and reflection are necessary. Dewey in 1933 already posed that “experience plus reflection equals learning”.

### 3.3 A system to support informal learning

How should a generic system be organised to support informal learning at the workplace? This question can be answered by enumerating the key knowledge and key abilities a system must possess:

1. The system should know the task the knowledge worker currently is working on as this to a large extent determines the learning context.

2. The system needs to define a learner’s learning goals in terms of the knowledge and skills that are required for performing each task.

3. The system should be able to identify a gap between the knowledge and skills that are required and the knowledge and skills that are already acquired.

4. The system should support the knowledge worker in acquiring the knowledge and skills to close a gap by supporting learning at the workplace by providing relevant material and the opportunity to communicate and collaborate with other knowledge workers while sharing the learning context.

5. The system should be able to find and present relevant material that fits the learning support and the knowledge domain the knowledge worker is active in.

The next chapters will indicate which challenges the project faced when designing such a system and which solutions were found.
4 Domain, Task, Learning Goal Model and their Relationships

This chapter explains the three model structures and their relationships which provide the basis for reasoning within APOSDLE:

- Domain model – provides a representation of the learning domain in OWL format
- Task model – provides a representation of the work tasks to be supported in YAWL format
- Learning goal model – provides a mapping between domain concepts, tasks and general learning goal types

We present the structures of the models, explain their roles (Sections 4.1, 4.2 and 4.3) and interdependencies (Section 4.4), and discuss the choice of modelling languages. In deliverable DVI.8 Application Partner Specific Models the reader will find the specific models created by the four application partners. These specific models utilize the structures presented here. In addition, in order to create these specific models the application partners employed the integrated modelling methodology. It was created in APOSDLE in order to streamline, simplify and support the modelling activities of the application partners and is documented in Ghidini et al., 2007.

4.1 Domain model

The objective of the domain model is to provide a semantic/logic description of the learning/work domain of an APOSDLE deployment environment. The domain is described in terms of concepts, relations, and objects that are relevant for this domain. Technically speaking the domain model is an ontology that defines a set of meaningful terms which are relevant for the domain and, which are used to classify and retrieve the knowledge artefacts (see also Chapter 7). Figure 4-1 provides an example of a domain model of APOSDLE (concepts, sub-concepts and properties) and shows it’s usage for the classification of knowledge artefacts.
In Figure 4-1, the upper part shows the domain model for one of the application partner's domains. The model is a hierarchical structure (left hand part) and contains relation types (right hand part). The red lines show how concepts and relations from this model are related to documents and knowledge artefacts.

4.1.1 Challenges

The research area of Knowledge Representation and reasoning provides a set of mature off-the-shelf approaches and tools for the representation of a domain. In APOSDLE we will build on top of these results, focusing more on the problem of generating a good representation of the domain rather than developing new representation paradigms. So the main research challenge in APOSDLE concerns
supporting the construction of appropriate domain models. The parameters that must be considered in order to build an appropriate model are the following:

- **Level of abstraction.** The same domain can be described at different levels of abstraction. Choosing the right level of abstraction is an essential issue to guarantee the effective usage of the domain model. If the level of abstraction is too high, then learning about generic concepts is in many cases useless. Furthermore, finding enough abstract description of a certain concept is quite often very difficult. At the opposite side, if the domain model is too detailed, then learning about one particular aspect could require one to learn about many irrelevant details. In defining the right level of abstraction one should take also into account the material available in the documents and the necessary effort for annotation. Again, a fine-granular domain description necessitates a detailed annotation of resources in order to make use of them. APOSDELE offers, via the associative network (see Section 7.4), the possibility of annotation at some intermediate level.

- **Coverage.** The domain model should completely describe all aspects of a learning domain.

- **Reusability.** The construction of a domain model is extremely expensive in terms of human resources. So it's important to build models that can be reused in many different situations.

- **Robustness to revision.** The models constructed should be robust for small changes and adaptations. Indeed, in the course of usage of the APOSDELE platform it will happen that the domain changes or it becomes necessary to represent a new part of the domain. This will require updating the domain model. Such an update should be done with minimum effort and minimal impact on other parts of the system.

- **Background knowledge.** The knowledge about a specific domain is typically composed of specific domain knowledge that is characteristic of the single domain, and some background knowledge, which is more general knowledge common to a large set of domains. For instance, in formalizing the RESCUE domain, the notion of an I* model is domain specific and concepts such as, user, application … are generic enough to be applicable to a wide range of domains. The research challenge here is to make available as many pre-compiled background knowledge as possible, and to allow a smooth integration with the domain specific knowledge.

### 4.1.2 The APOSDELE Approach

We based our representation on semantic web technologies. Domain models are mainly represented using OWL language (Web Ontology Language) which is based on Description Logics (Baader et. al, 2003). This approach allows expressing classes, properties and instances and axioms among them. This is standard methodology of the semantic web.

### 4.2 Task model

#### 4.2.1 Challenges

The research challenge described here is the process of finding a generic task representation language appropriate for the APOSDELE approach of workplace-integrated learning. Therefore we give a brief overview about the history of Business Process Management and Workflow Languages. Furthermore we try to motivate the use of the advanced workflow language YAWL with pattern support within APOSDELE and describe the APOSDELE task model.
4.2.2 Theoretical Background

4.2.2.1 Business Process Definition and Business Process Management

"Business Process" is a common term in modern business applications. Zur Muehlen (2004) interprets its meaning as a "high level process within a company, which contains activities interfacing with market partners". Whereas Gadatsch implies the use of information technology as a necessary aspect of a business process (Gadatsch 2002), other authors focus on its aims or targets like adding value (Johannson et al., 1993), strategic goals (Nordsieck, 1972) or customer benefits (Jablonski et al., 1997). The Workflow Management Coalition (1999) defines a business process as "a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.” Closely related to business processes, is the generic concept of "Business Process Management" (BPM): to manage business processes from their model design, via their implementation up to the controlling and optimization. Three main waves that mark the development and stages of BPM can be identified according to (Smith et al., 2003).

In the 1920s, Frederick Taylor’s theory of management dominated the area of BPM focusing on “methods and procedures analyses”. At that time, processes were implicit included in work practices. Process analysis and automation as happen nowadays, were not within the scope of Taylor’s approach. In the past decades, globalization and emerging markets led to stronger competition among the enterprises. BPR (Business Process Reengineering) as well as BPA (Business Process Automation) can mainly be characterized by the establishment of enterprise software applications like ERP or CRM. However, as a main characteristic of the second wave, efforts had to be conducted “at once”. Once implemented, the processes were hard to change on-the-fly by business managers. The third wave of BPM has started in recent years. The primary design goal is the facilitation of change. In contrast to the predominant approaches of the second wave, processes are created, administrated and changed on basis of process models. Thus it does not aim at one-off revolutionary changes to business processes, but at their continuous evolution. This is achieved by a synthesis and extension of several concepts, technologies and methodologies. BPM is composed of a number of concepts including workflow management, enterprise application integration, and many more.

4.2.2.2 Workflows and Workflow Languages

Modelling tasks and the associated context is a challenge since they usually form a complex setting depending on many relationships and constraints. Since task modelling descends from areas in HCI, most existing methods spring from this area. Depending on their purpose, the models vary in their degree of formality. Rather informal models like Hierarchical Task Analysis (HTA) (Annett, 2003) are used for analysis and concept development. HTA models are easy to understand; furthermore they can be used as communication artefacts between different stakeholders during systems design. Other models like GOMS (John & Kieras, 1996) and CTT (Mori et al., 2002) are more formal. Their formalisms make them machine readable but unusable for any work with non-experts in task modelling. The latter models have also been used by the model based user interface development community to (semi-)automatically construct user interfaces based on task models. Both usability and expressiveness have not yet been achieved in one single model. Existing models have all been developed with only one single purpose in mind. Theoretically, it would be possible to achieve both design goals in a single model. A human is only interested in a model's graphical representation, whereas the system only understands the formal specification. The challenge is to find suitable representations for both parties and a mapping between these two. In any case, it is clear that modelling tasks in a dynamic way entails the notion of workflows.

When talking about workflows, three main different perspectives have to be distinguished. As described in (Jablonski & Bussler, 1996), a workflow relates to a control-flow, dataflow and resource dimension. Since 1993, the WfMC\(^4\) is establishing standards related to workflow management. It is

\(^4\) Workflow Management Coalition
because of this cooperation of more than 300 manufacturers, scientists, users, and consultants that a widely accepted terminology framework exists today. According to Holey et al. (2004) and the Workflow Management Coalition (1999), a workflow is the automation of a business process. While a business process can be described by linguistic means only from a high-level view, a workflow has to be defined more specifically with technical respects. Workflow systems aim at providing an implementation tool for complex, recurring business processes. A variety of distinct features is offered by commercial workflow management systems. The Workflow Management Coalition (Hollingsworth, 1995; Workflow Management Coalition, 2002) and the Object Management Group (Object Management Group, 2002) attempt to provide standards for workflow management systems based on a multitude of workflow engines with different features. Several candidates have been considered from the areas of process modelling and general systems design including Petri-Nets (van der Aalst, 1996; van der Aalst, 1998), Event-Driven Process Chains (EPCs) (Scheer, 2000) and UML Activity Diagrams (Dumas & ter Hofstede, 2001), although none of these have achieved broad usage in workflow modelling. One of the most acknowledged procedure models for business process modeling is the ARIS method originating from Scheer (Scheer, 1998). According to this paper, it is a proprietary method based on the use of event-driven process-chains. ADONIS is a more generic approach that permits the use of several different methodologies (like UML). It is provided by the BOC-group (BOC group, 2007). The recent Workflow Patterns initiative (van der Aalst et al., 2003) has taken an empirical approach to identify the major control constructs that are inherent in workflow systems through a broad survey of process modelling languages and software offerings. Many of these requirements recur quite often during the requirements analysis for workflow systems.

Abstracting those requirements serve as a useful basis of key components for workflow languages. The definition of patterns for categorizing recurring problems and solutions in a particular domain is generally attributed to Christopher Alexander (Alexander et al., 1977). The concept of a patterns language implies a description of the dependencies and coherence between specific patterns. A patterns-based approach towards the identification of generic workflow constructs was first proposed by van der Aalst, (2002), which identified several control flow patterns relevant to the control flow perspective of workflow systems. Van der Aalst et al. (2003) have identified a number of workflow control patterns which specify the range of encountered control flow constructs while modelling and analyzing workflow scenarios. The authors describe a series of workflow patterns that aim to capture the diverse ways of workflows’ representations and utilizations. By defining those patterns in a workflow technologies and language independent way, they are able to provide a comprehensive treatment of the workflow data perspective. Russell (2006) subsequently expanded it to encompass twenty control flow patterns together with a formalization and analysis of their implementation in fifteen commercial and research workflow systems.

In addition to workflow control patterns, workflow data patterns aim to capture the various ways in which data is represented and utilized in workflows (Russell et al., 2004a). By defining those patterns in a technological and language independent way, the authors are able to provide a comprehensive management and handling of the workflow data. Workflow Resource Patterns (Russell et al., 2004b) try to capture the diverse ways of representing and utilizing resources in workflows. A comprehensive treatment of the resource perspective is given by the resource patterns.

All the introduced patterns build a basis for a detailed comparison of a number of commercially available workflow management systems and business process modelling languages. Furthermore, those workflow patterns have provided the conceptual basis for the YAWL workflow system (van der Aalst & ter Hofstede, 2005). YAWL is an acronym for “Yet Another Workflow Language”, an initiative that aims to provide both a workflow language that is based on the workflow patterns and also an open-source reference implementation that demonstrates the manner in which these constructs can interoperate. YAWL is a very expressive Petri-net based workflow description language with support for a broad range of the workflow patterns currently proposed by the workflow patterns initiative. It fuses the advantages of strictly formal semantics and reduced syntactic modelling complexity.
4.2.3 APOSIDE Approach

Crucial requirements for selecting a Workflow Management language in the APOSIDE context are:

1. **Easy and intuitive graphical notation:** Many project partners are involved in the APOSIDE project. Not all of them have a solid IT-background. It is therefore very important for the success of a potentially usable workflow modelling language not to require extensive training. For example, one should be able to understand the semantics of notation symbols without reading pages of documentation.

2. **Full formal specification:** The language should be clearly specified and not allow for ambiguities. This lack of formality is, for example, the case for certain aspects of UML, where different interpretations are sometimes possible.

3. **High degree of expressiveness:** The higher the expressive power of a language, the lower the "effort needed to construct models that reflect the process logic in a direct manner" (van der Aalst, 2002). First this aims at the general ability of languages to realize certain control-flow constructs (like, for example, loops), second it involves the resulting complexity of models if, for example, a workaround has to be used.

4. **Available Tools and Environment:** Software applications such as supportive graphical modelling tools should be available. As the workflow management system should be productively used and integrated into the IT landscape of the APOSIDE learning environment, the presence of a functioning environment for (amongst others) administration and enactment of workflows is also a prerequisite.

5. **Extensibility:** The Workflow Management System serves as a future basis for modelling and running processes within the APOSIDE framework. Since this interplay with the learning environment belongs to the subject of research, the possibility to view and eventually manipulate internal functionality of the Workflow Management System might be necessary. An open-source workflow management system is preferable for this reason.

After those needs were defined, several workflow modelling languages/systems were analyzed. A decision matrix (see Table 4-1) was created in order to balance reasons for or against the selection of a particular product. The evaluation included tools using the well-known concepts of Petri-nets, OMG’s Unified Modelling Language, OMG’s Business Process Modelling Notation (BPMN), IDS Scheer’s Event Driven Process Chains, the relatively unconventional approach of Concurrent Task Trees (CTT) and finally YAWL. We can see that YAWL seems to best fit the specified requirements.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Petri Nets</th>
<th>BPMN</th>
<th>UML</th>
<th>EPC</th>
<th>CTT</th>
<th>YAWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Easy and intuitive graphical notation</td>
<td>○</td>
<td>○</td>
<td>-</td>
<td>○</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>(2) Full formal specification</td>
<td>+</td>
<td>○</td>
<td>-</td>
<td>○</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>(3) High Degree of Expressiveness</td>
<td>-</td>
<td>+</td>
<td>○</td>
<td>○</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>(4) Available Tools and Environment</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>(5) Low cost</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>(6) Flexibility</td>
<td>○</td>
<td>○</td>
<td>-</td>
<td>-</td>
<td>○</td>
<td>+</td>
</tr>
</tbody>
</table>

* accomplished | ○ partly accomplished | - not accomplished

Table 4-1: Decision Matrix for the Workflow Management System Selection

The “Yet Another Workflow Language”-approach was chosen as a description language for representing the knowledge worker’s work processes for several reasons. First, YAWL employs strict formal semantics which eases the use for formal workflow interpretation and workflow mining. Second, it provides workflow patterns (see Section 4.2.2) which are easy-to-use abstractions for complex workflow constructs. And third, a number of tools for the YAWL workflow language are already available. The YAWL workflow language and the YAWL process editor are easily adoptable to the APOSIDE-specific requirements.
Figure 4-2 depicts the general APOSDLE task model which is a modified subset of the YAWL task model.

![Figure 4-2: APOSDLE Task Model](image)

4.2.4 Where else has something like this been used

A main impulse that drove the development of the YAWL-language was an evaluation of the expressiveness of several commercial and non-commercial workflow modelling languages by Van der Aalst et al. (2003). The focus of this evaluation was on the control-flow. As a criterion for expressiveness, so-called "workflow patterns" (see also Section 4.2.2) had been identified in advance, then it was tried to realize the patterns using each language. The evaluation revealed obstacles of different magnitudes regarding the realization of the patterns. In some cases, patterns could generally be realized, but used workarounds unnecessarily blew up the resulting visual representations and made them more difficult to interpret. There were also situations where it was not even possible at all to cope with several patterns using a specific language. To summarize, a certain lack of expressiveness within all examined languages was found. This led to the development and use of YAWL as described in Van der Aalst et al. (2003).

4.2.5 Next challenges

The short-term activity that followed this work is the development of task models which are focused on the particular work domains of the application partners (see for results Ghidini et al., 2007). Being able to properly employ the (modified) YAWL workflow management concepts, it will next be possible to further drive the APOSDLE learning environment as a whole into a productively usable state. Thinking about the long-term, the YAWL workflow would have to be embedded into a supportive software application landscape. This would comprise, for example, the linking of the workflow to a requirements database or graphical applications for editing related context models.
4.3 Learning goal model: a competence-based approach for formalizing learning goals and prerequisite knowledge

4.3.1 Challenges for realising adaptivity in work-integrated learning with APOSDLE

One of the main goals of APOSDLE is to provide a user with learning content in a highly adaptive manner. For the selection of learning content, the APOSDLE environment should take into account the actual learning need of a knowledge worker. The actual learning need shall be determined by requirements of a task at hand, and by a knowledge worker’s existing knowledge and skills, which has originated from previous learning and working experiences.

For realising this kind of adaptivity, two types of models have to be realized. First, information about a user in terms of his or her existing knowledge and skills needs to be stored. In adaptive learning research, the model in which this kind of information is maintained has been called the learner model (also student model, see Murray, 1999; Albert et al., 2002). To realize adaptivity to a full extent, the learner model has to be updated according to the user’s learning progress while engaged in the use of the system. In case of APOSDLE, the learner model is stored as part of the user profile (see section 3), and the update is one of the services performed there. Second, in order to adapt the provided learning content to the tasks that have to be performed in the work process, a second model needs to define a learner’s learning goals in terms of the knowledge and skills that are required for performing each task. This second model is the learning goal model and provides a mapping between the task model (see section 4.2) and the knowledge and skills needed to perform the tasks (learning goals).

Looking for a common conceptual basis for these two models, we chose a competence-based approach in the context of APOSDLE to address the challenges mentioned above. The use of competencies has often been advocated as a way to deal with the challenges in workplace learning (Green, 1999; Lucia & Lepsinger, 1999) for several reasons. Competencies are being used to more closely relate learning to organizational requirements such as organizational goals, or task requirements. Putting personal competencies in the centre of professional education seems necessary as the content of tasks is changing so rapidly that requirements can not be defined in detail. The shift to competencies is therefore not a fashionable hype, but a necessity for organizations to cope with uncertainty (Weinert, 1999).

Another reason why competencies have received significant attention is technical in nature. The fact that competencies can be used as an abstraction to describe several different types of objects (such as persons, tasks, content objects and others) with the same vocabulary, make them a good candidate for reducing the complexity by introducing a semantic layer (Hockemeyer et al. 2003; Sicilia, 2005). However, this use of the concept has lead to a diversity of definitions and conceptualizations discussed in more detail below.

Though the concept of competency is of research interest in several different scientific disciplines (for example, organizational psychology, educational sciences, management), the term lacks a standard definition. In the management literature, for instance, competency can be a trait of an individual, a group or an organization (see, for example, Rumsey, 1997). A related definition speaks of “core competencies” (Prahalad & Hamel, 1990), and refers to the unique capabilities of a business organization to deliver products and services, giving it its competitive edge on the market. A rather psychological definition has been brought up by Boyatzis (1982), who defines a competency as “an underlying characteristic of a person in that it may be a motive, trait, skill, aspect of one’s self image or social role, or a body of knowledge which he or she uses” (p. 21). This is rather typical for the way competencies have been conceptualized in the area of job analysis where a major effort has been undertaken to describe job requirements in standard terms (for example, as KSAOs, see below) in order to establish a common language for all Human Resource processes (like recruiting, selection, training and appraisal) (Green, 1999; Lucia & Lepsinger, 1999). In the educational sector, competencies are seen as a way to focus curriculum development more on generalizable proficiencies and abilities rather than on specific and isolated skills (van Assche, 2007). Finally, existing conceptualizations of competencies have been taken up in the Information Systems field either to
support existing HR Information Systems (Schmidt & Kunzmann, 2006) or in the area of eLearning (Sicilia, 2007).

Not surprisingly, this diversity in the adoption of the concept of competencies has lead to a plethora of definitions. In a review, Weinert (1999) concludes that “in all of these disciplines, competency is interpreted as a roughly specialized system of individual and/or collective abilities, proficiencies, or skills that are necessary or sufficient to reach a specific goal” (p. 4).

These conceptual problems come along with several requirements on modelling learning for APOSDLE. In order to allow for adaptive learner support, the model needs to have an interface with the user profile present in the learning environment. Furthermore, in order to allow for adaptive support with regard to a task at hand, there has to be a link to the task model of the APOSDLE environment. As content has to be retrieved that takes into account the learning need of a user, an association with the domain ontology is required. Predicting a worker’s performance in a “new” task requires inferences about prerequisite relationships among tasks. Computing optimised learning paths requires prerequisite relationships among different learning goals to be achieved.

4.3.2 The conceptualizing of learning goals in APOSDLE

Looking at the requirements of a competence-based approach for APOSDLE, it became clear that a much narrower conceptualization would suffice, and hence a narrower scope should be devised for the model. First, as the main target is to detect a knowledge worker’s learning need for a task at hand, and devise appropriate learning goals, the concept of a learning goal was established as the central concept around which adaptivity should be realized. A learning goal is understood as a discrete unit of knowledge or skill which a knowledge worker can attain at a certain point in time by performing learning activities.

Second, the scope of learning in APOSDLE is limited to learning at work, which also determines the scope of the learning goals modelled in APOSDLE. Which ones to include and exclude has to be fine tuned with the learning support provided (see Section 6.1). Also it has been shown (see 1st Workplace Learning Study, de Hoog et al., 2006) that learning events should be concise and cover a time range of minutes to hours rather than days to weeks, which again is an indication of a finer granularity. A further limit on the scope of learning is an emphasis within APOSDLE on domain specific learning activities, rather than generic (soft) skills.

As we are seeking a competence-based approach for devising and structuring learning goals, we are next reviewing a number of conceptualizations of competencies from which we then derive a conception of learning goals for APOSDLE.

Because of their great popularity, various approaches have been brought up to categorise and organize work-related competencies. For instance, New (1996) proposed a “three-tier model” of managerial competencies. By means of a case study, van den Berg (1998) suggested to classify competencies into Knowledge, Behavioural Style, Cognitive Capacity, and Personality. Within another conceptualization, the KSAO approach (see for example, Schippman et al., 2000; Lievens et al., 2004) led to categorizing competencies into Knowledge, Skills, Abilities, and Other characteristics. The KSAO classification has also been used for structuring competencies in the o*net database on occupational information (National O*NET Consortium, 2005).

All these approaches have in common their high degree of abstraction, which is not desirable in the APOSDLE context for the abovementioned reasons. Since learning artefacts are retrieved by means of a domain ontology, the abstract concepts of the competency model would have to be linked to concrete domain model elements, which constitutes an additional potential source of errors for modelling. One way to overcome these difficulties has been suggested, for instance, by Sánchez-Alonso & Frosch-Wilke (2007). The authors present a way for integrating competency-related concepts in domain ontologies. In APOSDLE, we want to go one step further. We argue that all learning goals, by definition, have to be related to the domain model. In other words, each element in the domain model is related to a (different) learning goal. However, an ontology in itself, when it only formalizes factual (or declarative) knowledge about a domain, is not sufficient for deriving learning
goals. Instead, learning goals refer to a cognitive action. They describe a state of a person in which he or she is able to do something: perform a task, solve a problem or perform some kind of mental operation. In short, learning goals need to describe the way declarative knowledge is used or applied. This is in line with conceptualizations which connect semantic models to a competency conceptualization. For example, Pernici et al. (2006) describe skills in terms of how domain knowledge is being used: a skill is a knowledge object plus an action verb.

Preferably, this second component of a learning goal (the action verb) is derived from a general instructional theory and should be independent of a specific learning/work domain. Anderson & Krathwohl (2001) have presented such a conceptualization which we adopt. We use the cognitive process dimensions of Anderson & Krathwohl (see for more details Section 6.1) remember, understand, apply, analyze, evaluate and create) to establish the procedural component of a learning goal which we will call its learning goal type. Taken together, the combination of one domain model element and one learning goal type constitutes one learning goal. This conceptualization is in line with recent developments in other FP6 projects, such as the iClass project (see, for example, Albert et al. 2007 or the Calibrate project (van Assche, 2007).

Summarizing: As we are aiming for a narrower scope in terms of the units of knowledge to be acquired, we have devised the concept of learning goal as the main unit of analysis. This will be detailed in the subsequent sections. In order to also be consistent with the broader competency conceptualizations mentioned in the review above, we will be seeking to link our conceptualization to these approaches. A first attempt in doing so, is presented in Section 4.3.7 where we compare our approach to standard industrial competency management practice.

4.3.3 Role of Learning Goals in APOSDLE

The role of learning goals within the APOSDLE approach is shown, in a simplified way, in Figure 4-3.

![Figure 4-3: The role of learning goals within the APOSDLE approach](image-url)

Performing a task is the main objective of a knowledge worker. This may lead to a learning need associated with the task he or she wants to perform. The task requires certain knowledge and skills
which are modelled as discrete learning goals. The mapping between tasks and learning goals is formalized in a Learning Goal Model. Given a valid mapping, and given a valid learning history in terms of learning goals the learner has engaged with in the past, APOSDELE can detect the learning need of the worker in terms of the learning goals he or she needs to attain. Each learning goal is related to an element in the domain model (domain concept), for each of which (ideally) learning content is available. This way, APOSDELE should be able to retrieve knowledge artefacts tailored to the learning need of a user, taking into account both the learning need as specified by the task at hand and the learning history of a worker.

More concretely, we are using the learning goal model for three purposes. First, we approximately infer a worker’s knowledge and skills from his or her past task engagements instead of an explicit (self-) assessment. This procedure is termed task-based learning history. Second, we are determining the learning goals of a user for a task at hand in terms of knowledge and skills he or she has to acquire by performing a learning need analysis. Third, we are computing an optimised sequence, a learning path, for attaining learning goals taking into account a prerequisite relation on the set of learning goals (this will be described in Section 6.1).

4.3.4 The learning goal model and its theoretical background: competence-based knowledge space theory

In order to compute a task-based learning history, and a learning need analysis, a formal model is needed that allows for inferences about what knowledge is required for a certain task, and which learning goals a user may potentially have when engaged in the task. Given such a model, conclusions can be drawn from a worker’s past task engagements about his or her most likely state of knowledge. Given the knowledge state of a worker and the knowledge requirements of a task at hand, a discrepancy can be identified and educational interventions can be initialised. Moreover, prerequisite relationships among learning goals need to be represented in the learning goal model in order to derive individualised learning paths.

One approach for dealing with prerequisite relationships among learning goals has been described in the context of the Intelligent Guide system (Khuwaja et al., 1996), a competency-based curriculum sequencing technology for educational purposes. In the Intelligent Guide system, domain concepts are linked via nodes in a network. The core of the system is the pedagogy engine, which is responsible for interpreting the knowledge state of the learner. Depending upon his or her state of interaction, it determines a learning path in order to attain a certain learning goal. The behaviour of the pedagogy engine depends on the domain knowledge in the knowledge network, the user’s actions, an assessment of the user’s knowledge state, and the pedagogy knowledge represented as rules in the engine.

Recently, several models for ontology-based learning systems were proposed for competence-based technology-enhanced workplace learning. For instance, Sicilia (2005a) introduces the basic elements of an ontology of competency, where prerequisite relationships between knowledge elements, attitudes, skills and competency elements are explicitly modelled. Similarly, Schmidt & Kunzmann (2006) propose to model competencies hierarchically decomposed into other (sub-) competencies (subsumption relation). Within the latter two conceptualizations, a user’s learning history in terms of knowledge and skills he or she has acquired is stored in a learner model (see, for example, Albert & Mori, 2001). In a concrete learning situation, the learner model has to be related to a model of learning goals for the learning domain. Moreover, besides the mapping between learning goals and their prerequisites, a mapping between learning goals and work tasks or job situations has to be established for taking into account the user’s work context. Otherwise, the suggested learning path would be independent from the task at hand. Taken together, a learning goal model needs to connect the working domain (tasks) with the learning domain (learning prerequisites).

Ley et al. (2005) have suggested Competence-based Knowledge Space Theory as a framework to formalise learning goals and their connection to workplace performance for work-integrated learning. With the Competence-based Knowledge Space Theory, Korossy (1997) has introduced an extension of Doignon & Falmagne’s Knowledge Space Theory, which has been developed in the 1980s and 90s.
as an attempt to model a person’s knowledge as close as possible to observable behaviour (Doignon & Falmagne, 1985, 1999). This was developed as a qualitative set-theoretical approach for the exact testing of a learner’s knowledge in a technology-enhanced learning environment. Knowledge Space Theory has been applied in adaptive testing and tutoring scenarios and systems (ALEKS Corp., 2003; Hockemeyer et al., 1998). One fundamental idea of Knowledge Space Theory is that a person’s knowledge in a certain domain can be described as the set of problems this person is able to solve. This set of problems constitutes the person’s performance state. The second fundamental assumption of Knowledge Space Theory is that prerequisite relationships exist between the problems. This comes along with two implications that are relevant for APOSDLE. First, not every combination of problems constitutes a feasible performance state. The collection of all feasible performance states is called a performance space. Second, from a person’s performance concerning one problem, it is possible to infer his or her performance in other problems.

In an attempt to develop Knowledge Space Theory further, Korossy suggested that in addition to the set of problems, one should look at the set of elementary competencies that are needed to solve the problem. This would generate information about the reasons for different levels of performance, and thereby help to suggest instructional interventions. Similar to the set of problems, elementary competencies are also structured in a competence space which results from a prerequisite relation on the set of competencies.

The relation between the two sets (problems and elementary competencies) is formalised by an interpretation function which maps each problem to a subset of competence states which are elements of the competence space. This subset of competence states contains all those competence states in each of which the problem is solvable. The competence state in which a person has to be minimally to solve a certain problem is termed the competency interpretation of a problem. The interpretation function induces a representation function which assigns to each of the competence states all problems which are solvable in that competence state. Which problems are solvable is determined by the interpretation function. If a knowledge domain is structured by a competence space, a performance space, an interpretation function, and a representation function, we call the model a competence-performance structure.

Competence-based Knowledge Space Theory has been applied in technology enhanced learning solutions. For example, Hockemeyer et al. (2003) assigned “competencies required” and “competencies taught” as metadata to a collection of learning objects. Thereby, prerequisite structures could be derived for the eLearning content which allow for adaptive tutoring. New course content could easily be integrated, as metadata was only held locally.

The learning goal model in APOSDLE - which could be termed as a task-learning goal structure - is formalized according to the principles of a competence performance structure in the following way. The elements of the two sets are the tasks the knowledge worker has to perform in the workplace and the learning goals which describe the knowledge and skills needed to perform these tasks. We assume both prerequisite relations on the set of tasks and on the set of learning goals. The mappings between the two sets are called interpretation and representation function. A feasible set of learning goals is called a knowledge state and it induces a task representation which gives all tasks that can be performed in that knowledge state.

A brief formal example for a task-learning goal structure in a knowledge domain encompassing the set of tasks \{T1, T2, T3, T4\} and the set of learning goals \{K1, K2, K3, K4\}\(^5\) can be found in Table 4-2. Table 4-2 shows the interpretation function, the representation function, the learning goal interpretation of each task, the feasible knowledge states, and the matching task representations of the example structure.

---

\(^5\)Note that usually there are more tasks than competencies in a learning domain.
Table 4-2: Interpretation and Representation Function of the Task-Learning Goal Structure (Example)

The column “Learning goal Interpretation” indicates for each task the knowledge state a person has to be minimally in to be able to perform the task. For example, in order to perform the task (T3), the knowledge state \{K1, K2, K3\} is required. Crosses specify the interpretation function in the table by assigning to each task all knowledge states in which the task is solvable. The task (T4), for instance, is only solvable by a person who is in knowledge state \{K1, K2, K3, K4\}. Task (T2) is solvable in knowledge states \{K3\}, \{K1, K3\}, \{K1, K2, K3\}, and \{K1, K2, K3, K4\}. The row “Task Representation” shows for each of the knowledge states all tasks that are solvable in that knowledge state. For example, a person who is in knowledge state \{K1, K3\} is able to solve the set \{T1,T2\} of tasks.

The prerequisite relation and a task-learning goal structure can be visualised by means of a Hasse-Diagram (Figure 4-4). Interpreting lines going from bottom to top lead to prerequisite relations. In the prerequisite relation on the set of learning goals, for instance, (K1) is a prerequisite for (K2) and (K4).

![Figure 4-4: Task-learning goal structure, prerequisite relations on the sets of tasks and learning goals](image-url)
Another advantage of Competence-based Knowledge Space Theory is that it incorporates a model of the knowledge domain and a model of the individual learner. Moreover, a task-learning goal structure can be combined with a domain ontology, and there are various possibilities for evaluating the validity of a task-learning goal structure in a learning domain.

### 4.3.5 Using the learning goal model in APOSDLE

As mentioned above, in the scope of APOSDLE, we use Competence-based Knowledge Space Theory for three practical purposes: for building a task-based learning history, for performing learning needs analysis, and for computing learning paths. Below these will be described in more detail.

**Building a task-based learning history.** Competency testing has a long tradition in business and organisational psychology, and several approaches were proposed for testing a worker’s work related knowledge and skills. Most of them require significant efforts and have a rather low validity, since very often the test situation and the test material have a moderate validity for predicting concrete task performance (see, for example, Arvey & Murphy, 1998). Our approach for assessment will be less ambitious. As the main purpose for building a learner model is to arrive at the best fitting learning goal for a specific task and a specific user, we will restrict ourselves to collecting a learning history based on tasks the user was engaged with in the past. The learning history for each user consists of a collection of learning goals which approximates the most likely knowledge state of the user (see for more details Chapter 5).

Due to the direct link of tasks and learning goals in Competence-based Knowledge Space Theory, there is an opportunity to approximately assess a worker’s minimally needed knowledge state by reviewing his or her past task performance. The rationale behind this approach is one that regards the application of knowledge in performing a task as a valuable learning experience. This type of workplace learning has been discussed as informal and experiential (see, for example, Garavan et al., 2002, but also Chapter 3).

For approximating the minimal knowledge state, we look at the set of tasks the worker has engaged with in his/her past use of the system. From the learning goal interpretation of each of these tasks, we infer this state. Several algorithms are conceivable for this inference. The easiest approach is to regard the minimal knowledge state of a person as the union of the learning goal interpretation of all tasks he or she has engaged with in the past. For instance, if a worker in the example of Table 4-2 has performed the tasks (T1) and (T3), the set \{K1, K2, K3\} would be regarded as his or her minimal knowledge state. The appropriateness of an algorithm for building a task-based learning history depends on the validity of the learning goal model and on other properties of the learning domain, such as the number of learning goals or the number of tasks.

**Performing learning needs analysis.** In order to provide the best-possible support for a worker who tackles a certain task, the worker’s learning need has to be specified. For this purpose, we employ a learning needs analysis in which the requirements of a task in terms of knowledge and skills are compared with the knowledge state of a worker. Again, several algorithms might be useful in this context. The easiest one would be to take the difference set between the learning goal interpretation of the “target” task, and the current knowledge state of the worker. In the example of Table 4-2, if the target task is (T4), and the set \{K1, K3\} constitutes the worker’s current knowledge state, the set \{K2, K4\} represents the worker’s learning need. Each element of the learning need then refers to one learning goal.

**Computing learning paths.** One of the main assumptions in technology enhanced learning derived from pedagogical principles and theory, is that the learning process can be improved through guidance (Schmidt, 2005). The concept of a prerequisite relation in the Competence-based Knowledge Space Theory has direct implications for the best-possible sequence for learning goals to be achieved. Returning to the example from above, if the set \{K2, K4\} constitutes the learning need of a learner, he or she should satisfy (or reach the associated knowledge state) (K2) before (K4), since (K2) is a prerequisite of (K4).
4.3.6 New challenges and ideas to solve them

Competence-based Knowledge Space theory has proved to be a promising approach for modelling learning goals and their relation with tasks and performance in APOSDLE, as evidenced by the 1st and 2nd prototypes. Nevertheless, new challenges did surface.

In order to improve the building of learning histories, a deterministic approach as described above may not be effective, especially because the model operates on data that involves a fair amount of randomness. Instead, several stochastic assessment procedures have been suggested in knowledge space theory (Falmagne & Doignon, 1988). In these approaches, a learner is not thought to be in one single knowledge state, but instead knowledge states are endowed with a probability distribution which describes the likelihood that the person has certain knowledge available. This would entail that in terms of Table 4-2 for example, the knowledge state \( \{K1\} \) has a probability value. Reasoning about knowledge states and prerequisites must be adjusted to take these probabilities into account.

The fact that we are concerned with learning implies the necessity to model and depict changes in the learner model (Sicilia, 2005b). On the one hand, this means to update the user profile according to the learning progress (see Chapter 5). Ideally, this update happens to a large extent automatically, as the learning environment detects the learner’s use of the system. Since in APOSDLE learning happens directly in the task context, there is an opportunity for updating the learner model according to past task executions (see Ley et al., 2006). On the other hand, up to now, we do not deal with “unlearning” or “forgetting”, that is, reducing the set of attained learning goals. Theories and algorithms should be sought in order to take the “decay” of knowledge into account (see for example, Bielikova & Nagy, 2006).

A further challenge would be to take into account qualitative levels of knowledge and skills in use. This has been suggested, for instance, by Dreyfus & Dreyfus (1986) and Eraut (1994) who identified five stages of skill development (Novice, Advanced Beginner, Competent, Proficient, and Expert). This would allow for a better tailoring of the learning material to the level of expertise of a learner.

As not only the accomplishment of a certain task could serve as “competence-indicating event” (see Won & Pipek, 2003), algorithms should be developed for making inferences about which knowledge and skills a worker has available by tracking his or her interaction(s) with the system. For instance, the participation in a chat about a certain topic, or the publication of a document could be such competence-indicating events.

4.3.7 Comparison of the APOSDLE Modelling Approach to Standard Industrial Competency Management Practice

4.3.7.1 Purpose of the Comparison

The purpose of this section is to compare the competence-based approach using learning goals as the unit of operation as used in APOSDLE to standard practice of competency management in industry settings. The need for such a comparison was prompted by the first APOSDLE review in which it was recommended to consider whether the use of competencies as the basis of discussion on employees’ learning needs fit into the employers existing use of competencies for job descriptions, personal learning plans, appraisal or career development.

This section makes a first step into this direction by comparing the use of competencies in APOSDLE to standard industrial practice, and by giving an outlook on the future developments in this important field of research.

4.3.7.2 Procedure

The following documentation was analyzed for performing the comparison: An excerpt from the deliverable “D6.1 Requirement analysis and specifications: Industrial use cases” as well as the
deliverable “D6.2 Industrial use case: Design and modelling of the specific metadata and models” both from the LUISA project and authored by Anne Monceaux and Joanna Guss from EADS-CCR.

4.3.7.3 Results

The analyzed documentation contains information about the conceptualization of the competence domain within the Airbus division at EADS with some concrete examples of positions and required competence profiles. The documentation also contains descriptions of a use case specifying how the approach is being applied at Airbus. Although being specific for Airbus, the approach taken there corresponds to several other industrial cases the author is aware of, such as those described in Wöls, Kirchpal & Ley (2003), Hiermann & Höfferer (2003) and Pernici et al. (2006). As such the EADS case can be seen as a good example of standard industrial practice in Competence Management.

<table>
<thead>
<tr>
<th>Dimensions considered in the analysis</th>
<th>APOSDLE Approach</th>
<th>EADS Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose of Modelling</td>
<td>Building a task-based learning history</td>
<td>Performing job based learning needs analysis</td>
</tr>
<tr>
<td></td>
<td>Performing task-based learning need analysis</td>
<td>Devising training measures</td>
</tr>
<tr>
<td></td>
<td>Devising learning paths as sequences of learning goals</td>
<td>Supporting competence appraisal in annual performance review sessions</td>
</tr>
<tr>
<td></td>
<td>Supporting work-integrated learning</td>
<td></td>
</tr>
<tr>
<td>Granularity of the concepts used</td>
<td>Learning Goals: Single elements of knowledge or skill needed to perform a task and which can be acquired in a work-integrated fashion. Knowledge state: A collection of knowledge and skills together with the ability to apply them in a number of tasks.</td>
<td>Knowledge &amp; Skills: Knowledge and know-how that need to be demonstrated for a competence</td>
</tr>
<tr>
<td></td>
<td>Competence: Ability to apply a number of activities, ability to apply knowledge and skill</td>
<td></td>
</tr>
<tr>
<td>Job vs. Task Based Analysis</td>
<td>The task performed in the workplace is the point of departure, irrespective of the position or job in which it is executed.</td>
<td>The job or “profession” is the point of departure in modelling and is used to model devise the learning need.</td>
</tr>
<tr>
<td>Separation of competence and performance</td>
<td>Clear conceptual differentiation between task performance and knowledge. Knowledge and skills can be utilized in a number of tasks.</td>
<td>One to one correspondence between activities and skills. Skills seen mainly as abilities to execute certain activities.</td>
</tr>
<tr>
<td>Proficiency Levels</td>
<td>Qualitative approach that takes into account application of knowledge in different contexts.</td>
<td>Employing a scale (1-5) for Competencies, as well as skills and knowledge, as an attempt to measure knowledge in a context free fashion.</td>
</tr>
</tbody>
</table>

Table 4-3: Overview of differences between the APOSDLE and the EADS approach

---

http://www.luisa-project.eu/www/
Six dimensions were considered in the comparison. The dimensions constitute important characteristics of modelling in general and for competency modelling in particular. They are not independent from each other. In each of the dimensions, marked differences in the approaches have been identified. Table 4-3 gives an overview of the differences found.

**Purpose of Modelling.** Clearly, the purpose of the APOSDLE approach is narrower in focus than a standard competency management approach. While work-integrated learning is only one aspect of standard HR processes, it also has certain special requirements, namely (1) that it is more fine grained in nature than in usual job based learning, and (2) that it is more strongly embedded and driven by task executions. These two issues will be covered in more detail below, as they explain two of the main differences found. Furthermore, the two last differences analyzed (3 and 4) relate to a more general difference in the philosophy and history of the two approaches. These will be discussed subsequently.

(1) **Granularity of learning objects and granularity of the concepts used.** A clear difference was found in the granularity of the targeted learning objects or knowledge artefacts. The finer granularity needed for work-integrated learning needs to be reflected by the granularity of the concepts used. As a result, the main unit of analysis in the APOSDLE approach is on the level of units of knowledge and skills, not encompassing the broader aspects of competencies. However, in line with research and practice, we see a competency composed of a collection of related units of knowledge and skills together with the ability to apply them in work-related tasks. In contrast to many approaches where the distinction between these two levels is not made explicitly, the APOSDLE approach has a very clear conceptual model to deal with this. A knowledge state (consisting of a collection of learning goals) and the associated task representation (consisting of a collection of tasks) can be seen as the equivalent to a competency in the sense described above.

The example given in Table 4-4 illustrates this by comparing one knowledge state from the APOSDLE learning goal model of Prototype 1 with a part of a position profile in the EADS Competency Model. In the EADS model, a competency requires a set of knowledge and skills. In the APOSDLE model, a knowledge state is composed of a set of learning goals as well as a set of associated tasks. The examples in the table are not equivalent, as different domains have been modelled. They are given here to illustrate the possible linkage of concepts used in standard HR competency management and in the APOSDLE approach.

(2) **Job vs. Task Based Analysis.** Standard HR practice is usually job based. This relates both to modelling (for example, bundling knowledge and skills and competencies in a job driven fashion), and to learning (for example, deriving competency gaps from a position profile). This is the practice reflected in the EADS model. Table 4-4 shows that the model has specified *professions* which entail concrete *positions* (which in turn can be held by particular persons). A profession consists of a consistent group of activities and is composed of competencies, skills and knowledge which are needed in that position. Position profiles can then be compared to employee profiles to devise optimal learning opportunities.

In APOSDLE, the point of departure is a task, irrespective of the position in which it is undertaken. In our view, the more dynamic organizational settings in knowledge work require that we employ a more flexible approach than a job based approach is able to provide. Of course, a grouping of fine level concepts (such as learning goals or knowledge & skills) to form more abstract concepts (such as competencies or knowledge states) is always possible. As we have demonstrated in the example above, such a grouping is not constrained by jobs or positions but can be formed more flexibly by grouping any collection of tasks in the task model to form a viable knowledge state. As in the EADS case a position is defined to encompass a set of activities (tasks), it is certainly possible to group those activities to form a position profile.
Table 4-4: Comparison between a knowledge state from the APOSDLE learning goal model of Prototype 1 and one part of a position profile in the EADS Competency Model

(3) Separation of competence and performance. Such a separation is not clearly foreseen in the EADS approach, where a “skill” is defined as the “ability to perform a particular task”, and hence defines a one-to-one correspondence. The approach we are following in APOSDLE allows us to keep track of the task context in which certain knowledge has been applied in the past. As a unit of knowledge (or learning goal) is mapped to a number of tasks, a learner model may keep track of the various tasks during which a learner has applied that knowledge before.

(4) Proficiency Levels. Similar to the previous point, the learning goal modelling in the APOSDLE approach follows a different philosophy than standard competency modelling practice. Knowledge Space Theory, as the underlying framework, follows a qualitative measurement approach and presents a sharp departure from traditional parametric measurement approaches in the social sciences (on which standard competency management practice is usually based). Again, an important point here is to express expertise in terms of the (qualitative) task context in which knowledge has been applied, rather than in terms of a context free measurement scale. This does not preclude that certain scales (like the ones mentioned in section 4.3.6) can not be used in the future. However, these would need to describe differences in expertise in qualitative ways rather than quantitatively. Furthermore as also mentioned in section 4.3.6, the APOSDLE approach is by no means committed to a deterministic approach.

4.3.7.4 Discussion and directions for future research

Summarizing the comparison: what we found were differences in terms of granularity (1) and the general organizational approach (2). We have shown a clear conceptual schema of how these two issues will be dealt with by aligning the APOSDLE discourse with that employed in standard competency management practice. It needs to be further researched if the alignment we are seeking
is actually feasible and beneficial. This would require finding ways to (a) align knowledge states and competencies, and to (b) align the job based approach with the task based approach. The EADS case within the APOSDLE project is a good opportunity to continue with this research and answer the questions raised above.

We have further found two differences which relate to the conceptual differentiation between competence and performance (3) and qualitative vs. quantitative measurement approaches (4). As these are the core of the underlying conceptual model, we will continue to research ways for integrating this into industrial practice.

4.4 APOSDLE knowledge base

The previous sections described the three models that are the core of APOSDLE, which can be labelled the APOSDLE Knowledge Base (AKB). Within this AKB each model is stored in its original format (OWL, YAWL, matrix) and also in the meta-model schema. The meta-model schema is an ontology represented in OWL. The advantage of keeping the models in both representations is that within the original representation (for example, YAWL for tasks) reasoning about processes is possible while within the OWL representation reasoning about the overall model is supported.

4.4.1 Challenges

The APOSDLE knowledge base is to provide a machine processable integrated theory (= knowledge base) of the application domain in which the APOSDLE platform supports E-learning. This implies that at every new installation of APOSDLE one needs to specify a new domain model, task model and learning goal model. APOSDLE provides for this a modelling methodology and a set of modelling tools that support the methodology.

- The first objective is to provide a methodology and a set of tools for acquiring knowledge for the AKB. To successfully achieve this objective, we have to support domain experts in eliciting their knowledge in a logical model without requiring them to be expert logicians. Furthermore, we have to exploit as much as possible those company resources, such as a list of key terms, classification schema, database schema, and other schemata, which are used by the companies to organize their document archive, data and activities. These schemata constitute the result of an analysis and a conceptualization of the domain, and encode domain knowledge in a quasi-logical format.

- The second objective is integrate in the AKB different types of knowledge, such as the procedural knowledge encoded in the process model, the knowledge about competences of workers, and knowledge about the domain models. The main difficulty here is to establish the connection among the different types of knowledge. For instance, stating that a certain task needs some competences and that a task will deal with a certain concept.

- The third goal is to provide a set of services on top of the AKB for the APOSDLE learning tool and the APOSDLE classification tool. These knowledge services should be able to answer both logical queries, that is, queries that imply some form of logical reasoning in the models, and non-logic based queries, that is, queries that implies some form of analogical, statistical, and similarity based reasoning on the knowledge. Again the problem is being able to reason about different, but related, logics (that is, the logic of processes, the logic of the competency model and the logic of the domain model).

To each of these challenges we devote a section.

4.4.2 Knowledge acquisition

The problem of knowledge elicitation has been around since the beginning of Artificial Intelligence. In the early years, to build expert system human experts were asked to express the knowledge about a
certain domain into machine processable languages such as rules. Recently, with the advent of the semantic web, we have seen a major shift on how knowledge should be expressed. In particular logic has become the basic tool to express machine processable knowledge, thanks to its declarative fashion, its unambiguous meaning and the availability of efficient reasoning tools. This shift makes the process of knowledge elicitation more complex than in the past, as the meaning of logical formulas and the effects of axioms is much more complex than simple activation rules. Domain experts, nowadays, need more support than in the past in the process of knowledge elicitation. Furthermore, the recent development of the web (in particular WEB 2.0) highlights the importance of the social aspects of knowledge. In our setting, this means that the elicitation of knowledge is not a process that is performed by a single individual in isolation, but is more a social negotiation process. In this negotiation process, the members of a community (the group of people who share and use a common piece of knowledge for their activity) should converge to a common representation of the shared knowledge.

In the area of knowledge engineering there are several well established methodologies and tools for knowledge elicitation. One of the best known methodologies is the CommonKADS methodology (Schreiber, et. al, 2002). This methodology was especially designed for building expert systems. Recently a large number of methodologies and tools has been developed for many aspects of the complex process of building and maintaining ontologies. For instance DILIGENT (Vrandecic, et al., 2005) is a methodology focusing on the evolution of ontologies and in particular on users and their usage of the ontology, instead of on the initial design, thus recognizing that knowledge is a tangible and moving target. HCOME (Kotis et al., 2006) supports the development of ontologies in a decentralized fashion, which in turn supports the creation of an agreed ontology between a group of people in a community. These are only some examples of knowledge acquisition methodologies. The paper by Cristani and Cuel (2005) provides a survey of proposals for ontology creation methodologies. All these tools, however, could not be simply adopted in the APOSDLE project, as these methodologies need the development of a set of intelligent tools that supports the automatic performance of their steps.

For the APOSDLE prototypes 1 and 2, a more traditional approach was used. A person knowledgeable about the domain constructed the domain model which was represented in the Protégé language. For prototype 2, the knowledgeable persons in each domain entered an informal version of the domain model into a Wiki template, which was automatically converted into a Protégé representation. However, the informal Wiki template can be used in the future as a means to make knowledge acquisition more into a community activity.

4.4.3 Knowledge Integration

The basic idea of knowledge integration in APOSDLE is based on federated, diverse knowledge bases. In this federation, each type of knowledge (knowledge related to working, knowledge related to learning, and knowledge related to the domain) is stored in its original format (the YAWL representation (see Section 4.2), learning goals format (see Section 4.3) and the OWL representation (See Section 4.1) and the integration is done separately in what we can define as the meta-model schema (see Figure 4-5).
The meta-model schema is an ontology represented in OWL. The meta-model schema is a formal representation of the APOSDLE meta-model, which forms the conceptual basis for integrating the federated knowledge bases. The conceptualization of this approach is shown in Figure 4-6.

The main idea expressed in Figure 4-6: The approach for knowledge integration in the AKB is that we have two copies of the same knowledge. The first one in its original format (the lowest levels in Figure 4-6) on which all the customized reasoning tools can be applied. The domain model manager supports ontological reasoning on OWL models describing the application domain. The task model manager supports reasoning about processes expressed in YAWL. The learning goal model manager supports the competency gap analysis.

A second copy of each model, expressed in OWL format, is managed by the AKB manager which is used for the integration with all the other models. The AKB is also charged with the synchronization between the two copies of each model. Below we briefly describe several issues that arise in model integration.
4.4.3.1 Learning goal model – Task model

As was defined in section 4.3 the notion of a learning goal is intrinsically connected with the ability to perform a certain task. That is, a task is an activity that requires the mastering of certain learning goals. The mapping between tasks and learning goals is a binary relation, which we denote with "requires" between tasks and learning goals. Formally:

\[ \text{Requires} \subseteq \text{Task} \times \text{LearningGoals} \]

This simple definition raises a number of representational questions:

- How does the specific structure of the task model and the domain model affect this relation? For instance, if task \( T \) is, say, “Define Simulation Objective and Requirements” and is composed four sub-tasks (for example, Identify and understand customer needs, Elaborate Simulation objectives, Specify Simulation Requirement and Specify Simulation Requirements), what is the relation between the learning goals of the sub-tasks and the more general task? The inheritance pattern, as present in the concept/sub concept relation, is not so trivial, as in some cases sub-tasks can be delegated and therefore one does not need the specific knowledge for the subtasks. In APOSDLE we have tackled this problem by disallowing any propagation of learning goals through tasks.

- A second issue concerns standard learning goals. Tasks can be clustered by abstract task-types. Examples of abstract task types are Acquiring, Eliciting, Monitoring, and Estimating. For each abstract task type one can think of a set of standard learning goals, so that any real life task of such a type, inherits the learning goals from the type to which it belongs. For instance, to perform a task of type “Formalising”, it is necessary to acquire knowledge of formal languages, which is the learning goal.

- Another interesting topic is the capability to automatically attach a subset of learning goals to a task on the basis of its description. For instance, the task “Elaborate Simulation objectives” should require, in any case, the knowledge of what is a simulation is and what is the objective of a simulation. This kind of mappings between learning goals and tasks could be discovered by analyzing the labels and the descriptions of the tasks.

4.4.3.2 Learning goal model – Domain model

The basic idea is that learning goals of the user should be linked to domain concepts (see Section 4.3.3). For instance, if in the domain model we have the concept of “interview” then the learning goals of a knowledge worker could be “being able to perform an interview” or “knowing what is an interview”. The idea in APOSDLE is that these learning goals can be pursued by extracting learning material from the knowledge artefact which is related to such a concept (see Section 6.1). The ontology itself, with its axioms, metadata and comments, can also be exploited for extracting useful learning material. For example, if a knowledge worker needs to know what is “Progressive Abstraktion” (example taken from the domain of a German partner) then, by accessing the domain model (s)he can obtain learning content like:

- A “progressive abstraction” is a “Kreativitätstechniken” (showing the concept of which it is a subconcept)
- A “progressive abstraction” is a method of systematically specifying problems. Assessing core questions of a problem with a systematic approach. (showing the definition of the concept)
- “Progressive abstraction” in English is Progressive Abstraction (showing it's translation into another language)
4.4.3.3 Task model – Domain model

The issue of combining procedural knowledge (described in the task model) with domain knowledge (described in the domain model) is not novel, but it is becoming a real issue in the web-services research area. However, there is not a standard approach to combine knowledge about processes (tasks) with knowledge about objects (concepts). The language OWL-S (Martin et al, 2004), is an extension of OWL for the specification of processes. Being an extension of the language for specifying APOSDLE processes, it could be a promising candidate for the integration of the process model with the domain model in APOSDLE. However, this activity is not yet planned in the project. An alternative language that integrates ontologies and processes is BPEL2WS (Business process execution language for web services). At the current stage more investigation is necessary in order to implement a real integration between task and domain model into one model.

4.4.4 Where else has something like this been used

Domain modelling via an ontology is a standardized methodology nowadays and most applications that support semantic information retrieval are based on ontologies. Similarly, the classification of documents (or any other set of object, like pictures, movies, etc.) with respect to an ontology is also a standard practice in the semantic web.
5 Context Awareness and User Profile Services

This chapter explains and stresses the importance of context awareness for eLearning systems in general and APOSDLE in particular. We start with providing the reader with the vision of context aware systems, their origins, and current developments in this field. Specifically a number of context aware systems are discussed. The APOSDLE approach to context awareness is based on a common architecture developed by Baldauf et al. (2007) which includes the use of agents in three layers separating the detection of context, planning and action based on context. It also details the challenges of user context determination. The goal is to identify a user’s current work task based on the user’s interactions with the system (key strokes, mouse movements, applications used, etc.) and the metadata and content of the resources accessed (mail messages, documents, links to people, etc.).

User context determination plays a crucial role within the overall APOSDLE approach. In order to be able to provide the user with information, learning material and links to people relevant to her task at hand, the system needs to identify the work task reliably. The identified task then updates the user profile (Section 5.3) and causes several activities to be started pro-actively: re-computation of a user’s learning goal and learning goal histories (Section 4.3), retrieval of knowledge artefacts relevant to the learning goal (Section 7.4), finding people relevant to the learning goal (Section 6.2), and the dynamic creation of learning events (Section 6.1). The results are displayed in a resource list and a people list.

APOSDLE also stores user related context information in digital user profiles. These profiles are used for maintaining the user’s usage history and current context with respect to their personal work-, learning- and collaboration-related experiences. The APOSDLE approach differentiates between four forms of user related data: user data, usage data, inferred data, and environment data. This layering of user profile information allows us to clearly separate between factual information and assumed information about the user. The component responsible for operations upon user profiles is the User Profile Service (UPS). The UPS’ functionality is made accessible to other parts of the APOSDLE system via a set of services. Examples are services for logging users’ activities and services computing queries for resources, which optimally match a user’s current learning need. The UPS’ contextualized services will be described (see Sections 5.2 and 5.3).

Finally, as APOSDLE will store quite some information about users, the issue of privacy is discussed (Section 5.4).

5.1 Context awareness and context determination

5.1.1 Challenges

In this Section the foundations behind the APOSDLE task detection engine, which is one of the major research challenges to face if you want to support unobtrusive real-time task-based context-aware learning, are explained. First, we introduce the concept of context awareness and explain possible context architectures and then we detail the context-based task detection engine in APOSDLE.

5.1.2 Theoretical background

The vision of ubiquitous computing and context-awareness was developed by Xerox PARC researcher Mark Weiser in the late eighties and early nineties of the previous century. In his landmark article “The Computer for the 21st Century” (Weiser, 1991) starts with the following sentences: “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it”. The terms context-awareness and context-aware applications were
introduced a few years after Weiser’s vision was published. Context-awareness was defined in several publications, most prominently in (Schilit et al., 1994; Brown, 1996; Abowd et al., 1997; Dey & Abowd, 1999; Schmidt et al., 1999a).

All definitions have in common that information from the users’ or the entities’ environment is acquired and used to trigger certain actions in the system. Using such context-information, systems adapt to a given situation to improve the user’s interaction with the system. Moreover, systems can trigger actions autonomously. Finally, context can be attached to information to make it easily retrievable. Once again, the use of context information in novel applications needs to be embedded in the way humans live. The earliest context-aware applications had a strong focus on location. Want et al. (1992) and Harter et al. (1999) presented the Active Badge system: it used ultrasound beacons to determine the position of active badges. By coupling the system to a telephone exchange, it allowed to route calls to the phone next to its recipient, even when the recipient was not sitting at his desk. The system was extended with mobile devices: the ParcTabs, an early PDA-like device (Schilit, 1995; Want et al., 1995). Brown (1996) presented stick-e notes: notes that are activated by context. As soon as a certain context, such as location, time, or others, becomes active, stick-e notes are delivered to the user. They envision the use of such notes for tourist guide applications, mobile workers, making surveys, and control panels (a note is activated close to the device that is to be controlled). Dey and Abowd (2000) described a similar system for context-aware reminders as well as many early projects, which focused on mobile or wearable devices.

Much work on the management of notifications can in fact be considered as context aware. Horvitz et al. (2003a), for example, presented the BestCom system that allows managing telephone calls in an office setting, depending on context factors such as, whether a conversation is taking place in the office or not. Microsoft’s ongoing MyLifeBits project (Gemmel et al., 2002) aims at collecting and storing any digital information about a person, but leaves the annotation to the user. People do forget things, even if they are deemed important. Czerwinski and Horvitz (2002) studied how (quickly) users forget important events in a desktop-based computing environment. They aimed at identifying important events automatically and providing memory prostheses for important computing events: Beyond Capture & Access, a context aware application can support a user in performing a specific task. Therefore, Dehn and Van Mulken (2000) defined user interface agents as follows: “Interface agents are computer programs that aid a user in accomplishing tasks carried out at the computer, such as sorting email, filtering information and scheduling meetings. These agents differ from conventional computer programs in that they can act autonomously on behalf of the user, that is, without requiring the user to enter a command or click a button whenever she wants the task to be carried out. In addition to autonomy, a characteristic of intelligent agents is their ability to perform tasks delegated to them in an intelligent, that is, context- and user-dependent way.” Current research is particularly strongly geared toward user interfaces in which communication between humans and a computer is partly mediated by this type of interface agents (as originally introduced by Maes, 1994).

Classic texts on user interface agents comprise a collection of key papers, see Cassell, 2000; Koda & Maes, 1996; Rist et al., 1997; Dehn & Van Mulken, 2000, for a thorough review of evaluation techniques for embodied conversation agents. Well-known examples of interface agents are intelligent tutoring systems and context sensitive help systems, such as Selker’s COACH (Selker, 1994) or the Remembrance Agent (Rhodes, 1996). The latter is an autonomous interface agent that reminds the user of relevant files stored on the user’s local disk, thus “remembering” relevant material that the user has already seen. This was later expanded (Rhodes & Stamer, 2000) to explicitly make a contextually aware associative memory, for instance, to make literature suggestions based on the browsing behaviour of a web-user. Liebermann and Selker (2000) explained the use of context in user interfaces by the following: “Each application can be thought of as establishing a context for user action. The application determines what actions are available to the user and what objects can be operated on. Leaving one application and entering another means changing contexts. The user gets a different set of actions and a different set of objects. Each tool works in only a single context and only when that particular application is active. Any communication of data between one application and another requires a stereotypical set of actions on the part of the user (copy, switch application, paste).” The problem is that keeping track of contexts and acting upon them can be tedious to manage in the real world, and software agents are used in this case to keep the interaction as simple as possible and
manage context for the user. The primary job of a user interface agent in such cases is thus to understand the intent of the user. In such systems, the user can use the interface without explicitly taking notice of the agent; the agent can display suggestions, or directly manipulate objects in the interface, based on context as implicit input collected from the user. After these first publications stressed the importance of agents to manage context, many layered context-aware architectures and frameworks have evolved during the last years.

For key research on these architectures we refer to Baldauf et al. (2007), who collected a large number of framework and middleware examples, and identified a common architecture in modern context-aware applications by design analysis (see Figure 5-1). They also stressed the use of agents in three layers or modules that separate the detection of context (perception), planning (reasoning), and action based on context. This separation of detecting and using context is mainly utilized to improve extensibility and reusability of systems. Blackboard systems (Engelmore & Morgan, 1988) are often applied in multi agents systems. Winograd (2001) contains several pointers as to what the blackboard approach’s characteristic weaknesses might be for use in agent-based context aware systems: since it is loosely coupled it pays a price in communication efficiency. On the other hand, ease of configuring, simplicity and robustness are its benefits.

Context can be seen as a restricted set of important information and constraints in a specific situation. The restriction is based on a goal aimed at. Wessner (2005) qualifies a computer system as context aware if it supports the users by providing relevant information or services taking into account context information for the selection process. Dey and Abowd (1999) underlined that the relevance of the regarded services or information is also strongly connected to the users’ current tasks.

5.1.3 APOSDLE approach

We base our approach on the Baldauf et al. (2007) architectural paradigm (see Figure 5-1), since this is the most widely used one in related work. For task detection we implement layer 1 and 2. Layer 3 is later used for recommending suitable learning resources with regard to the current work task of the learner.

**Figure 5-1: 3-Layer Context Architecture (from Baldauf et al., 2007)**

In order to support the recommendation of suitable learning resources and many other aspects of an interwoven learning paradigm, the e-learning system needs to be aware of a user’s current work task. This information can be seen as a prerequisite for finding suitable resources or collaboration experts and is to be retrieved automatically and unobtrusively, using low-level context information as
indicators. The applicability of machine learning (ML) algorithms to this problem is an APOSDLE research question. The goal of task prediction is to know the active task of the user at any point in time. A task is defined as a unit of work consisting of activities to reach a certain goal. The problem of task prediction is seen as a machine learning problem. When first using the prediction system, it is untrained and the user needs to specify the task s/he works on from a predefined list of business tasks (by manual selection). During the work process a context monitoring component logs any desktop events reflecting the user’s actions. These include keyboard presses, application launches, full texts in documents, etc. As soon as enough classified events are gathered, the system trains a ML model of the user’s work task in the application domain. The optimal result is achieved when the user continues to work and s/he does not need to manually notify the system of task switches anymore. The task predictor automatically classifies the active tasks using continuously recorded event streams (automated selection). Whenever the classification engine detects a change in tasks, the e-learning environment displays, when needed for attaining a missing learning goal, a new list of associated learning resources and suitable experts relevant for the detected work task.

5.1.4 New challenges and ideas

The Prototype 2 user studies will address the evaluation of the APOSDLE task prediction in two ways. First, from a quantitative point of view: By comparing the predicted task with the real-world task set by the user, it is possible to calculate a quantitative measure of prediction accuracy. Second, from a qualitative point of view: By asking the users about their personal impression of the benefits of task prediction. A questionnaire will contain questions about the individually experienced quality and frequency of the APOSDLE task prediction. This allows a glimpse on the overall usefulness of the APOSDLE task prediction, whose real-world evaluation is limited by the constrained nature and the domain-dependency of the user studies.

Preliminary experiments show low recall for infrequent tasks and especially low precision for tasks at a high granularity level (that is, very general tasks like “Working on X”). Future research efforts are dedicated to increase both the precision and the recall of the APOSDLE task prediction. More elaborated methods for feature engineering and aggregation will be tested for this purpose.

Another research challenge is the evolution of the task prediction model. In order to support changing work processes we need to build incremental and continuously learning algorithms that take descriptive task executions by knowledge worker into account to improve the prediction model.

In the, rather unlikely, case the task prediction is widely accepted by the user and yields high quality measures, we can think about reducing the initial training effort which is necessary to generate a classification model.

5.2 User context representation

APOSDLE stores user related context information in digital user profiles. These profiles are used for maintaining the user’s usage history and current context with respect to their personal work-, learning- and collaboration-related experiences. APOSDLE uses data stored in the user profiles for adapting its offerings to the users’ needs and requirements. Based on the user profile, data recommendations are computed aiming at supporting the users’ learning goals attainment (See Section 4.3), the preparation of the retrieval of resources (see Section 7.4) and acts of collaboration (see Section 6.2). The component responsible for operations on user profiles is the User Profile Service (UPS). The UPS’ functionality is made accessible to other parts of the APOSDLE system via a set of services. Examples are services for logging users’ activities and services computing queries for resources, which optimally meet a user’s current learning need. Storage and manipulation of user related information, of course, raises critical issues concerning privacy and ethics in general. These issues are tackled by giving users large control over their own personal data. This Section will describe how the APOSDLE UPS deals with digital user profiles of knowledge workers. Additionally, the UPS’ contextualized services will be described and issues related to privacy will be discussed.
5.2.1 Challenge
Digital user profiles are used to represent personal and work-related properties of knowledge workers. The major challenge we are facing is to identify appropriate and sufficient user-related properties for a user's digital profile. The question to answer therefore is whether the representation chosen for digital user profiles is complete and consistent with respect to the domain in which it is used. We will present the conceptual and theoretical background behind our decisions and development in section 5.2.2. During development a number of new challenges emerged. Section 5.3.3 describes these new challenges and how they will be tackled in future prototypes.

5.2.2 Theoretical background and APOSDLE approach

5.2.2.1 Properties of APOSDLE user profiles
The theoretical background of our work with respect to user profile properties is influenced by two streams of research. The first stream is research on user context on a general level (see, for example, Dey et al., 2001, Kobsa et al., 2001, Oppermann, 2005, Wessner, 2005), the second is research on user modelling (see, for example, Kay 1995, Fink & Kobsa 2002, Fink 2004, Heckmann et al., 2005, Heckmann 2005, Howes & Smith, 2006, but also FOAF7, IEEE PAPI Learner8, IMS LIP9 or UbisWorld10). User context research traditionally differentiates between three forms of user related data: user data, usage data and environment data (see Kobsa et al., 2001). User data comprises data, which is related to a user personally. Examples are record data (for example, name, and address), characteristics such as gender, age and income, users' knowledge and previous experiences but also the users' preferences and goals. Usage data comprises all forms of users' activities, which can be observed such as selective actions, ratings, confirmatory actions (for example, purchase, print, save) or usage regularities. Environment data is usually made up of data related to the software environment (for example, operating system, browser and other applications used), the hardware environment (for example, display device) and the user's location.

The APOSDLE approach is based on the classification outlined above but adapts it slightly. First, we aim at arriving at a clearer distinction between factual information and assumed information about the user. Second, we aim at providing a sound basis for identifying those forms of user activities, which shall be managed by the UPS in the form of usage data. The APOSDLE approach differentiates between environment data, user data, inferred data and usage data. The respective semantics of these types of data are presented below.

For the APOSDLE UPS, environment data is mainly made up of influences, which are external to the user, but still have an impact on the user's work execution. For the APOSDLE User Profile Service the environment is the user's computational work environment plus the supporting infrastructure. The essential parts of the environment are modelled formally and models are mapped onto each other where this is feasible. The models and mappings are abstracted through the APOSDLE meta-model. The meta-model forms the conceptual basis for the environment data-part of the APOSDLE user profiles (Ulbrich et al., 2006). The meta-model is directly related to the models for the domain knowledge, tasks and competencies (that have been described in Chapter 4) and the mappings defined between them. The meta-model is formally represented as an ontology in the form of the meta-model schema (see section 4.4.3).

In APOSDLE user data is understood to consist of facts about the user (for example, name, address, contact details etc.), whereas more uncertain information is referred to as inferred data. An example of uncertain information is data, which represents a user's goal with respect to the future attainment of

---

7 http://xmlns.com/foaf/spec/
8 http://edutool.com/papi/
9 http://www.imsglobal.org/profiles/index.html
10 http://www.ubisworld.org/
learning goals. Inferred data allows finding ways of dealing with heuristics and assumptions. User data on the other hand comprises all known facts about a user, which only change slowly over time, if they change at all. Examples include date of birth, name, organisational unit and so forth.

*Usage data* comprises information coming from observations of the user’s behaviour during work execution. Examples for this are users’ previous work experiences or past collaboration with others. It is crucial, to make a clear distinction between forms of usage, which are relevant to the domain in question (that is, work-integrated learning) and usage which has no meaning for the domain. As a rule of thumb, those forms of usage are considered relevant which are directly related to model elements of the environment data. Usage data, as we envision it, thus reside on a more abstract level than on the level of operating system events such as mere keystrokes or mouse clicks (Budzik, 2000; Budzik, 2001; Maus, 2001; Rath et al., 2007; Lokaiczyk et al., 2007). Usage data is of great importance since from it other data (i.e. informed assumptions about the user) can be inferred. For future development it is planned to gradually incorporate more and more types of usage data (for example, what documents did the user read or write). The key criterion for deciding upon which information will be stored as usage information, are the user requirements. That is: if there is a requirement to store and later analyse a particular type of usage data, it will be taken up and stored within the User Profiles.

Figure 5-2 gives a brief, graphical summary of our taxonomy of types of user profile data. In this figure the user profile data is arranged in layers. The innermost layer consists of the factual personal information (user data). The second layer (usage data) covers, what can be observed about a user during her work execution. The third layer (inferred data) is computed from observations stored in the usage data. The outermost layer (environment data) is actually not directly related to the user profile as such nor is it stored or maintained within the UPS. Nevertheless, environment data has a certain impact on the user profiles and therefore needs to be considered.

![User Profile Data](image)

**Figure 5-2: User profile data arranged in layers as implemented for the APOSDLE user profile**

Table 5-1 shows in detail the pieces of information, which are stored in user profiles. For each piece of information it is pointed out, how it is classified with respect to the taxonomy of user profile data, which has been introduced above.
<table>
<thead>
<tr>
<th>Type of data</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>User data</td>
<td>Authentication information</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>Affiliation (organisational unit)</td>
</tr>
<tr>
<td></td>
<td>Contact information (mailbox)</td>
</tr>
<tr>
<td></td>
<td>Gender</td>
</tr>
<tr>
<td></td>
<td>Date of birth</td>
</tr>
<tr>
<td></td>
<td>URL of a picture of the user</td>
</tr>
<tr>
<td>Usage data</td>
<td>History of tasks executed in the past</td>
</tr>
<tr>
<td></td>
<td>History of learning goals attained in the past</td>
</tr>
<tr>
<td></td>
<td>History of collaborations the user has participated in</td>
</tr>
<tr>
<td></td>
<td>History of learning events the user engaged in (not implemented yet)</td>
</tr>
<tr>
<td>Inferred data</td>
<td>Knowledge and skills acquired through successful task execution</td>
</tr>
<tr>
<td></td>
<td>A ranking value, which indicates the degree of ‘expertise’ of a user with respect to a given learning goal</td>
</tr>
<tr>
<td></td>
<td>A ranking value, which indicates the degree of ‘connectedness’ of a user with respect to a given other user</td>
</tr>
<tr>
<td>Environment data</td>
<td>Representation of task model, learning goal model, domain model, collaborations and organisational unit</td>
</tr>
<tr>
<td></td>
<td>Mapping between task model and learning goal model</td>
</tr>
<tr>
<td></td>
<td>Mapping between learning goal model and domain model</td>
</tr>
<tr>
<td></td>
<td>Mapping between task model and a set of YAWL IDs (see section 4.2)</td>
</tr>
</tbody>
</table>

Table 5-1: Services offered by the APOSDEL UPS

We chose this approach because it provides us with a clear distinction between factual knowledge, observations, mere assumptions and influences from the user’s environment. Each type of data can be treated in such a way, that its specific characteristics are sufficiently respected. Factual user-related data is stored within a directory service such as LDAP\(^\text{11}\). Observations about a user are stored as lists of entries reflecting a user’s usage history. Inferred assumptions are not stored as such, but computed on demand. Environment data is stored in domain-dependent models outside the UPS. This allows for greater flexibility in adding new properties to the user profiles in the future. If, for instance, the need would arise to add the property “organisational unit” to the user data it would instantly be clear where to store it (LDAP) and how to maintain it.

Similar approaches have been proven to work, especially in the domain of recommender systems (Montaner et al., 2003) the most well known example being of course Amazon\(^\text{12}\). Amazon stores for each user personal information (user data), a history of purchased and rated items (usage data), recommendations (inferred data) and additionally holds a huge taxonomy of its products and services (environment data). The APOSDEL approach lacks the ratings (although we plan to incorporate it, see section 5.3.3), but extends the Amazon model in that the UPS considers more dimensions than mere

\(^{11}\) http://tools.ietf.org/html/rfc4510

\(^{12}\) http://www.amazon.com
purchases. In the domain of educational systems, ELM-ART (Brusilovsky et al., 1996) and especially ELM-ART II (Weber & Brusilovsky, 2001) follow a similar approach. Users are modelled using different layers, where all layers are basically consistent with APOSDLE’s usage data and inferred data: ELM-ART II keeps track of visited learning units and learned units (usage data) and known and inferred units (inferred data). Another educational system is KBS Hyperbook (Henze & Nejdl, 2001). Here the user model is basically constructed from concepts of a knowledge base (environment data). User models are fed into a Bayesian network, which calculates probabilities of a user’s knowledge state with respect to a given concept (inferred data).

5.2.3 New challenges

It has turned out that a new challenge emerged during designing and developing the first prototype of the UPS. This new challenge is mainly concerned with adaptations of existing user profile data based on users’ relevance feedback.

For the time being, the APOSDLE UPS relies on a rather small set of information sources, from which information about users and the users’ preferences can be acquired. Moreover, the information, which is acquired, is ‘static’ in the sense that it merely covers pre-modelled environment data. As a consequence, it is currently hardly possible to make a statement about a user, which goes beyond pre-modelled and pre-assigned knowledge (or assumptions, respectively). Further sources of information about users, are required. According to Hijikata (2004) there are two ways to obtain information about a user and a user’s interests. The first way is to let the user explicitly enter information (through, for example, ratings, questionnaires etc.), the second way is to analyse the users’ natural behaviour and infer implicit interest indicators from that (see. Kelly & Teevan, 2003 and Köberl, 2006). For the second prototype we plan to focus more strongly on acquiring and assessing both forms of user relevance feedback. We will evaluate user acceptance of the outcomes of the UPS’ inference and production services and the results of the retrieval service (see section 7.4). Results of the evaluation will then serve to update user profile data or to improve other services’ outcomes.

5.3 User profile services

The component responsible for operations upon the user profiles described in the previous section is the User Profile Service (UPS). The UPS’ functionality is made accessible to other parts of the APOSDLE system via a set of services. Examples are services for logging users’ activities and services computing queries for resources, which optimally match a user’s current learning need.

5.3.1 Challenge

The UPS makes the contents of the user profile and results from computations upon them available to the other parts of the APOSDLE system. These other parts are, for instance, the associative retrieval mechanism (see section 7.4), the process assembling learning events (see section 6.1) or the application facilitating collaboration and communication (see section 6.2). Data from user profiles needs to be analysed and made available by the UPS in such a way that other parts of APOSDLE can adapt their own contribution to the users’ current needs in order to support learning in the best possible way. The challenge we are facing here is to come to a clear conceptual understanding of how the UPS makes its functionality accessible to other parts of the APOSDLE system. The approach we chose rests on services. Section 5.3.2 describes our approach. New challenges which appeared during development are described in section 5.3.3.

5.3.2 Theoretical background and APOSDLE approach

The theoretical background of our work basically stems from research on systems that are designed to collect, process and make available user profile data through a set of user-focused services. A service is crafted towards an actual, concrete need of a knowledge worker or an application supporting the
knowledge worker. The selection and design of services has been informed by research in recommender systems (Montaner et al., 2003, Middleton et al., 2004, Adomavicius et al., 2005, Adomavicius & Tuzhilin 2005), user context lifecycles (Kobsa et al., 2001, Teltzrow & Kobsa, 2004), knowledge management (see, for example., Maurer & Tochtermann, 2002), the semantic desktop (see, for example, Sauermann et al., 2006) and social semantic desktop (see, for example, Decker & Frank, 2004). According to research on recommender systems (Montaner et al., 2003) and the user context lifecycle (Kobsa et al., 2001) user-focused services are divided into services for acquiring user related data (acquisition or profile generation and maintenance), inferring further information from it (secondary inference or profile exploitation) and producing a result, which is to be presented to the user (production). Kay and Kummerfeld (2006) additionally suggest including further services, which allow for larger user control over their own data (control).

The APOSDLE approach rests upon the approaches illustrated above. Logging services basically acquire usage data (see Section 5.2.2.1). Generally speaking this type of services is accessed by the APOSDLE Tools directly. An example for a rather simple logging service is the service which collects a user’s current task need: an APOSDLE Tool detects the task (see Section 5.1), which a user is currently working on. The Tool then sends this information to the APOSDLE Platform. The Platform passes the information on to the User Profile Service. The User Profile Service’s logging mechanism relates the incoming task-related information to the appropriate User Profile. Additionally, more complex logging services exist. These services are usually made up of a number of simpler logging services. An example is how invitations for a collaboration session are managed. An invitation consists of a two-way handshake between two users: the first user invites the second one. The second one accepts or rejects the invitation. Once an invitation is accepted the collaboration is considered to be established. This information goes into the user profiles of the inviter and the invitee. As indicated above, usage data is stored in a history-based way as a list of entries. The responsibility for detecting information, which can then serve as usage data, lies upon APOSDLE’s client-side tools.

Inference services operate upon usage data. Generally speaking, inference services examine usage histories and analyse the users’ past behaviours. Based on this analysis, characteristics are computed, which are meaningful in the user’s current situation. An example for such a service is the computation of the learning goals, given the current task of the user. This service first computes the user’s learning history (that is, which knowledge and skills has a user acquired in the past). Next, the service computes the gap between the learning goals that must have been attained to carry out the task at hand and the user’s learning history. And finally, the learning goals are ranked according to the (assumed) optimal sequence in which they shall be learnt. We believe that with inference services new knowledge about the user can be constructed based on observations of the user (see Maurer & Tochtermann, 2002).

Production services and control services are more conventional by nature than inference services. The UPS produces for example a list of tasks a user has executed in the past. Some production services exploit mappings defined between models of the environment data. An example for this is a service detecting all domain model elements which have been mapped onto a given learning goal. For the time being, only two control services are implemented: deletion of tasks and deletion of learning goals from a user’ profile. These services are directly invoked by users themselves. Figure 5-3 is an illustration of the UPS’ services. Logging as well as control services expose their interfaces to the APOSDLE tools and allow users to input data. Production services expose their interfaces also to APOSDLE tools, but merely serve for presentation purposes. Inference services are only used for computations, which are internal to the APOSDLE UPS.
Table 5-2 shows a comprehensive list of the UPS’ services.

<table>
<thead>
<tr>
<th>Type of service</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition</td>
<td>User login</td>
</tr>
<tr>
<td>(logging)</td>
<td>User logout</td>
</tr>
<tr>
<td></td>
<td>Successful task execution (not implemented yet)</td>
</tr>
<tr>
<td></td>
<td>Current task need</td>
</tr>
<tr>
<td></td>
<td>Current learning need</td>
</tr>
<tr>
<td></td>
<td>Management of invitations</td>
</tr>
<tr>
<td>Inference</td>
<td>Learning goals assessment</td>
</tr>
<tr>
<td></td>
<td>Learning need</td>
</tr>
<tr>
<td></td>
<td>Learning goals ranking</td>
</tr>
<tr>
<td></td>
<td>Expertise with respect to a given learning goal</td>
</tr>
<tr>
<td></td>
<td>Connectedness with respect to past collaborations</td>
</tr>
<tr>
<td>Production</td>
<td>User data as name value pairs</td>
</tr>
<tr>
<td></td>
<td>Current domain model elements</td>
</tr>
<tr>
<td></td>
<td>Ranked list of learning goals</td>
</tr>
<tr>
<td></td>
<td>List of experts</td>
</tr>
<tr>
<td></td>
<td>List of past collaborators (‘peers’)</td>
</tr>
<tr>
<td></td>
<td>History of successfully executed tasks</td>
</tr>
<tr>
<td></td>
<td>History of learning goals attained through task execution</td>
</tr>
<tr>
<td></td>
<td>History of collaborations</td>
</tr>
</tbody>
</table>
We chose this approach because we pursue the goal of providing a user-focused emergent system. Emergent means that the UPS not only collects data but also offers advanced computations and inferences over the data collected. From a conceptual viewpoint, for this inference services are required. Offering acquisition services too, is a rather natural consequence, given that data, which is to be processed, needs to be acquired at some point in time. Production services for presenting external actors with results from computations and control services for user control round off the classification of service types. Logically, the separation between acquisition services, inference services, production services and control services appears to be plausible.

Similar approaches have been proven to work in knowledge management (KM) on a conceptual level. Maurer and Tochtermann (2002) have introduced a model of KM systems. This model defines functionalities of KM systems and classifies these functionalities. The model differentiates between acquisition services, which rely on explicit input of information and those, which rely on implicit input. Additionally, production services are subdivided into services, producing results on demand (for example, based on an explicit query) and services, producing results autonomously. The model also makes a difference between inference services creating “new knowledge” based on observations of user behaviour as opposed to services creating “new knowledge” based on existing knowledge. Compared to the APOSDLE UPS, this approach lacks the control services. Further similar approaches have been proven to work in recommender systems (Montaner et al. 2003, Perugini et al. 2004). Systems like GroupLens, Ringo/Firefly, LikeMinds, (again) Amazon and others, infer similarities between users from users’ past behaviour. Nevertheless, for the time being, our research has revealed that the separation of service types and the classification of actual services according to service types are new and original contributions of the APOSDLE UPS. We did not identify further approaches, which chose to follow this approach to the same extent as the UPS.

5.3.3 New challenges

There are new challenges, which have emerged during the first phase of development. These challenges became apparent when it was analysed, which types of inference services were actually needed in order to meet users’ real-world use cases and demands. Below there is a brief outlook on ideas how to approach these issues.

Learning paths

For prototype 1 an algorithm for computing the learning goals has been implemented as an inference service (see section 5.3.2). The algorithm implemented, consists of three different steps: assessment of a user’s learning goals, computation of the learning need and ranking of learning goals with respect to the assumed sequence in which they should be learnt. For prototype 2 we plan to enhance the algorithm with respect to each of the three steps. We will especially take into account prerequisite relationships between learning goals as they are established through knowledge space theory (see Section 4.3 and Korossy, 1997). There are also plans to analyse users’ relevance feedback regarding

<table>
<thead>
<tr>
<th>Type of service</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>Deletion of tasks successfully executed</td>
</tr>
<tr>
<td></td>
<td>Deletion of learning goals attained in the past</td>
</tr>
</tbody>
</table>

Table 5-2: Services offered by the APOSDLE UPS

13 In this context Gruber (2007) writes about the differentiation between ‘collected intelligence’ and ‘collective intelligence’.
14 http://www.grouplens.org/ - a news recommender system
15 A music recommender system now owned by Microsoft
16 A clickstream analysis system now owned by Macromedia
the suggested learning goals ranking. As a consequence of evaluating user feedback, the personal profile of a given user will be updated.

Social network analysis

Social network analysis offers powerful methods to explore social relationship and social structures hidden within the data gathered by the APOSDLE system. It has already been applied in prototype 1 to calculate a list of peers presented to a user in the sidebar. The algorithm used for computing peers was rather simple and straightforward and did not take any hidden information about users' mutual relationships into account. To gather more detailed information about hidden relationships, we will incorporate different methods of social network analysis as, for example, described in Wassermann & Faust (1994) or Carrington et al. (2005). Our plan is to combine several of these results (for example, reachability, centrality, distance of users) in order to gain more realistic data for services provided by the APOSDLE UPS. Results from standard network analysis methods will therefore be used as a foundation for further improved calculations. Focus points in this area of work will be on collaborations between users of the APOSDLE system and on expert finding. Collaborations will provide means for two different groups of services. The first group of services deals with presenting users with collaboration partners, who are matching best with the users' current situation and environment. The second group of services provides users with information about their collaborations (for example, who have asked them and who they have asked for e.g. advices to a particular topic). For the task of expert finding, our aim is to enrich existing algorithms of prototype 1 with methods of social network analysis to include further aspects besides the set of learning goals the user has achieved in the past.

5.4 Privacy

A third major challenge is that legitimate concerns related to privacy issues need to be considered. For the first prototype of APOSDLE, this topic has been omitted on purpose. The second prototype will have to take care of privacy issues. The challenge is to give users large control over their own personal data and still offer comprehensive services, which focus on the users' needs.

The theoretical background of our approach, rests on several international guidelines, which have been published in the recent past aiming at helping system designers and developers in safeguarding that privacy issues are sufficiently considered (see, for example. European Parliament, 1995, OECD, 2002 and U.S. Department of Commerce, 2000). Based on these guidelines, we plan to implement privacy policies for the second prototype of the APOSDLE UPS. We plan to introduce privacy levels (see Clarke, 2006) as part of these policies. Users can choose whether they want to make themselves known to the system (“identified” user) or remain unknown. Identified users can further define whether they want others to have access to their user profile data (“identified public”) or not (“identified private”). Unknown users can decide whether they are “pseudonymous” (that is, no user identifier is related to stored pieces of information but a timestamp is kept) and “anonymous” (that is, no user identifier and no timestamp is kept). The main goal is to give users large control over their personal data and the way it is collected. Examples for this user control are: experts will be given a way to temporarily ‘hide’ their expertise from others (that is, if they are very busy and do not want to be distracted from their own work); users will generally be allowed to delete (parts of) their own usage data; users will be allowed to permanently make their usage data anonymous etc.

The APOSDLE approach, which is currently under development, consists of two mutually dependent steps. First, privacy levels are introduced into the UPS such that each user can select her preferred level of privacy. Second, an additional technical component has been designed, which is dedicated towards making sure that privacy policies are sufficiently considered. This component has been labelled Privacy Enhancement Service (PES). The PES reads policy files, which have been developed specifically for an organisation, and ensures that no single piece of user-related information is made available to third parties if they are not explicitly allowed to view this information. In doing this, the PES does not “reinvent the wheel” but utilizes an existing technological framework (eXtensible Access Control Markup Language, XACML) and adopts it to its needs. This framework was originally
developed by Sun Microsystems\textsuperscript{17} and further developed by Organization for the Advancement of Structured Information Standards (OASIS)\textsuperscript{18} and the Swedish Institute of Computer Science\textsuperscript{19}.

Similar approaches have been proven to work in several research projects (for example, Chadwick & Otenko, 2002 and Lorch et al., 2003). Generally speaking these projects are mainly concerned with Identity and Access management (IAM) that is, controlling a persons’ access to resources. It was pointed out that the IAM can greatly benefit from approaches similar to ours. This is especially true for systems, which need to control access to resources on a very fine-grained level (Vullings & Dalziel, 2005).

\textsuperscript{17} \url{http://sunxacml.sourceforge.net/}
\textsuperscript{18} \url{http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf}
\textsuperscript{19} \url{http://www.sics.se/spot/xacml_3_0.html}
6 Work Integrated Learning Support

In this chapter we will explain how APOSDLE will provide integrated support for the three roles a knowledge worker fulfills at the professional workplace: the role of worker, the role of learner, and the role of expert. Support is offered in several ways. First of all, by giving the worker access to (parts of) documents that are relevant for the task at hand and for the learning goals that can be derived from the task and the learning history of the worker. Second, by embedding these (parts of) documents in a context that should support active engagement with the materials, as well as reflective thought. Thirdly, by facilitating collaboration and communication with co-workers that have expertise with the task at hand or that have (mastered) the same learning goals. Section 6.1 deals with the first two ways to support learning, Section 6.2 is devoted to collaboration and communication.

6.1 Supporting learning

Presenting the right information at the right time can help a knowledge worker that has problems in performing a specific task, to complete this task, but this does not necessarily mean that (s)he has also learned something. For learning to take place the user has to actively process the information presented and give meaning, adjust existing cognitive models or develop new models or schemata. This idea is stressed in the instructional design theories of Merrill (2001) who states that an instructional strategy has at least two phases: presentation/demonstration and application/practice. In this Section we will elaborate on this and describe how the APOSDLE system can support learning.

6.1.1 Challenges

Instructional design generally starts with an analysis of learning needs which leads to a specification of learning goals. Next, the level of prerequisite knowledge is determined and, based on this, learning materials are selected and activities are specified that should enable the learner to reach the learning goals. These activities are evaluated to give the learner feedback on his learning process.

The challenge we face is that the APOSDLE system should be applicable across different domains. This means that beforehand the topics are not known and, furthermore, the types of material that are available are also unknown. Therefore it is impossible to generate specific (sequences of) instructional events (based on instructional design models) that should lead to a specific learning goal. However, we still have to create a situation in which a person is stimulated to learn something. The solution could be to embed parts of existing documents in a general context that contains elements to support learning. These elements are based on the conditions for learning as formulated by Gagne (see for instance Gagne, 1985).

6.1.2 Background and similar problems

Gagne described nine events that provide the necessary conditions for learning and serve as the basis for designing instruction and selecting appropriate media. These nine events are:

- gaining attention
- informing learners of the objective
- stimulating recall of prior learning
- presenting the stimulus
- providing learning guidance
- eliciting performance
providing feedback
assessing performance
enhancing retention and transfer

Since the content of materials that are available is unknown, we can only focus on these conditions for learning. This can be done by creating learning templates that address these conditions and that have slots that can be filled with material specific for a certain domain and learning need.

An example of such an approach in which templates are used, can be found in the work of Chacon (2003) who designed a method of object-oriented instructional design for Web based learning. The conceptual components of his approach are depicted in Figure 6-1. In his view, learning templates are like a container that can be filled with information (in any format) that is relevant to the course being taught. The information must be adapted to pedagogical rules that increase the likelihood of learning. These small units are called learning objects: small stand alone units of instruction that can be tagged with descriptors and stored for reuse in various instructional contexts (see also Hamel & Ryan-Jones, 2001). Each learning object consists of a content component and a strategy component. The content component refers to the questions “What?” and “Why?”; while the strategy component deals with the “How?”. One could see learning objects as a kind of LEGO blocks that are combined to build a course. However this LEGO metaphor is not appropriate because this implies that every block can be combined with every other block. This is not true when creating educational materials. In this setting some blocks may be required to base other blocks on and sequencing rules have to be applied.

An important difference with the APOSDLE context is that the objects in Chacon’s method were purposefully designed for instruction while in APOSDLE, in most cases, they are not, because they are retrieved from heterogeneous company document repositories (see also Section 7.4). The objects in APOSDLE only have a content component and lack a strategy component. The challenge is to develop generic templates that combine a content and a strategy component.

![Figure 6-1: Learning template as used by Chacon (2003)](image-url)
6.1.3 APOSDELE approach

In the APOSDELE approach are learning templates used to specify general goals which are linked to general types of learning materials and generic types of activities. From these learning templates learning events are generated automatically.

The general learning goals are based on the taxonomy developed by Anderson and Krathwohl (2001). In this taxonomy (see also Section 4.3) learning goals are categorized based on two dimensions: type of knowledge and cognitive processes. We use the cognitive processes dimension to specify general learning goals. The general processes we use are: Remember, Understand, Apply, Evaluate, and Create. These processes match the learning goals model, as described in Section 4.3.

To make the learning goals more specific we use sub-processes specified by Anderson and Krathwohl (see Table 6-1).

<table>
<thead>
<tr>
<th>General processes</th>
<th>Sub processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remember</td>
<td>Recall, Recognize</td>
</tr>
<tr>
<td>Understand</td>
<td>Interpret, Exemplify, Classify, Summarize, Infer, Compare, Explain</td>
</tr>
<tr>
<td>Apply</td>
<td>Execute, Implement</td>
</tr>
<tr>
<td>Evaluate</td>
<td>Check, Critique</td>
</tr>
<tr>
<td>Create</td>
<td>Generate, Plan, Construct</td>
</tr>
</tbody>
</table>

Table 6-1: Processes and sub processes as discerned by Anderson & Krathwohl (2001).

These sub-processes are not only used to make the goals more specific, but also to determine what types of generic learning activities can be specified. For instance, a meaningful learning activity related to the sub-process “compare” is to detect correspondences and differences between two instances of an example or a procedure.

Although the content is unknown, we need to have some clues as to what type of content should be selected by the APOSDELE system to fill a learning template and create a learning event, since not every combination of learning goals, activities and learning material will be meaningful. Furthermore, the chunks of information that are presented in a template should not be too large because this can cause a cognitive processing overload or information overload that can impede learning. Therefore, we need a generic categorisation of learning materials that could be used to specify and limit the type of content that should be presented in a learning template. There are some categorisations available (based on projects like LOM, Ariadne, SCORM), but these are rather low in instructional meaning. We have chosen a categorisation that is derived from the IMAT project (De Hoog et al., 2002). IMAT stands for Integrating Manuals and Training. This was a EU funded project in which fragments extracted from maintenance manuals, were classified in categories like: definition, overview, example,
assignment, guideline, how-to, summary, etc. These categories reflect the instructional role(s), or material use, a piece of information can play in an instructional process.

To construct the learning templates, meaningful combinations of learning activities (based on sub-processes) and types of material use were specified. Each meaningful combination was specified in a learning template. Each learning template has the same general structure (see Figure 6-2 for an example of a filled learning template (a learning event)) consisting of a header, a content section, an engagement activity with references to experts and or colleagues and evaluation questions. Social interaction and collaboration are important activities for learning because in these activities people have to express and explicate ideas, discuss, give arguments and reflect on actions. Furthermore, in social interaction other workers have the opportunity to give feedback, coach others and give additional information not presented in the learning events. Therefore the learning templates should contain elements that stimulate learners to engage in social interaction and to locate co-workers that have the same learning goal or experts that have already mastered the learning goal to which the learning event is related. The evaluation questions refer to different components of the learning template (event). The question about information elements is intended to find out whether the presented content was helpful for learning. In the second prototype these questions are different and a question is added that intends to subjectively establish whether the user has learned what she wanted to learn.²⁰

²⁰ The issue how to establish if the user has learned something is a thorny one. In Section 5.3.2 the notion of using success in task execution was discussed. In earlier versions of Learning Event examples, “exercises” were used, but they turned out to be not very well suited to a work context. Apart from the conceptual problem of creating generic tests (like the engagement activity in Figure 6-2), this issue will be investigated in more depth in the later phases of the project.
Competency: Knowledge of validating the SD model. Important domain elements are: SD_Model.

Activity: In this activity you are asked to judge and critique whether an explanation is appropriate based on certain criteria.

Id: 364

Learning content

Below a page of text with an explanation will be displayed.

```
Note:
In the SD, I tried to convert task and resources to soft goal or goals when possible.
I tried to minimize the number of dependencies in the SD Model and elaborate more on
the SR.

Assumptions:
The list of unpaid vehicles registration numbers are automatically displayed in the Capital
staff screen.
The Capital staff have to locate the required video image and then access it.
The list of paid vehicles is actually within the computer system. Thus the computer
system doesn’t depend on anything else when it tries to distinguish between paid and
unpaid vehicles.
The Computer system can’t achieve its goals unless it receives the video images from the
data network.
Capital are in charge of training the staff instead of TPL.
```

Engagement activity

Below you will find a list with criteria. Check the appropriateness of the explanation based on these criteria.

- Completeness
- Clarity
- Contradictory

Discuss your opinion with one of your colleagues.

Please select a partner to discuss with:

Coworkers:
- Sara Bourne (offline)
- Andrew Dorey (offline)

Experts:
- Bill Murray (offline)
- Andrew Dorey (offline)
- Philippe Maurice (offline)

Evaluation questions

Please tell us if this learning event helped you to learn (1 is disagree and 5 is agree):

- Information element(s): 1 2 3 4 5
- Engagement activity: 1 2 3 4 5
- Total learning event: 1 2 3 4 5

If you have any remarks, you can enter them here

Figure 6-2: General structure of a learning template (as used in prototype 1)
The upper part in Figure 6-2 specifies the learning goal that the learning template refers to and prerequisite knowledge (these are dynamic fields to be filled by the system) and an introduction to the learning activity. This part refers to the first three learning conditions (see Section 6.1.2): gaining attention, informing the learner of the objective and stimulating recall of prior knowledge. The content section is filled by specific types of retrieved knowledge artefacts (dynamic field, see also Section 7.4) and is related to the fourth condition “presenting stimulus material”. The engagement activity specifies the learning activity (fixed field) and when necessary a knowledge artefact can also be used in this activity field (dynamic field). Furthermore, an input field is part of this activity field. This input field is used to record reactions to the learning activity. This relates to conditions 5 and 6: providing learning guidance and eliciting performance. The bottom part of the template consists of references to experts and/or colleagues available and can also be used to ask the learner to evaluate the learning event. In the current templates the last three learning conditions (providing feedback, assessing performance and enhancing retention and transfer) are used when creating a learning event and are not shown in Figure 6-2. This is mainly because assessment and feedback are difficult to implement because the content that has to be evaluated is not known beforehand (see also section 6.1.4.).

To be able to fill these templates, parts of documents that are available do not only have to be labelled or tagged based on the learning goals that they refer to, but also based on the categorization of knowledge artefacts. Furthermore, a decision has to be made how large the knowledge chunk will be that will be presented to the user. When only the specific artefact is displayed, the context in which it is presented will be lost and this could mean that users have difficulties in valuating the meaning of the artefact. If the artefact is put into a large chunk there is the risk that the user is not able to identify the right part of the information presented and experiences a processing overload. In the first prototype the page on which the knowledge artefact can be found is presented.

A learning tool should be in place to fill the generic templates with artefacts that are retrieved from the documents that have been tagged to create run-time a learning event. A schematic representation of such a tool is given in Figure 6-3.

![Figure 6-3: Learning tool overview](image)

When several instances of an artefact are available that satisfy the criteria specified in the template, it is possible to generate more than one learning event based on one template. This means that a learner can choose which of the learning events (s)he wants to access.

The contents in a learning domain are not independent. One important relationship between concepts is the prerequisite relationship (see also Section 4.3.5). Prerequisite knowledge is knowledge that must be acquired before other knowledge can be learned. So, if learners lack prerequisite knowledge,
they should master that knowledge first. The knowledge that users bring to a learning situation also has a considerable impact on the learning process (Jonassen & Grabowski, 1994). Ausubel et al. state it more explicitly: “if we had to reduce all of educational psychology to just one principle, we would say this: The most important single factor influencing learning is what the learner already knows. Ascertain this and teach him accordingly.” (Ausubel, Novak, & Hanesian, 1968, p. 163). There are several explanations for the importance of prior knowledge. According to Ausubel et al., learning is an active process in which new knowledge is integrated in existing cognitive structures. Piaget speaks of assimilation of new knowledge in the existing knowledge structures and accommodation of existing knowledge structures to make place for new information. Learning in which new information is integrated into existing structures is called meaningful learning, because the new information is somehow related to existing knowledge. This is different from rote learning. In rote learning new knowledge is not integrated in the existing structures, but it is stored isolated from those structures. Therefore, knowledge gained through rote learning is easily forgotten, while knowledge gained through meaningful learning is remembered better. The activation of prior knowledge positively influences learning (Merrill, 2002). However, existing knowledge can also hinder the learning process. For example, incomplete or false prior knowledge can distort new information (Andre, 1997; Novak, 2002) and it is hard to change the existing structures in the learners’ mind. In the second prototype of the Learning Tool we want to address these issues by providing an overview of the contents and the relationships between the concepts in a domain. That way, learners in APOSDLE will get a better understanding of the concepts and their relationships.

APOSDLE stores information about the domain knowledge and the learner to identify learning paths that are tailored to the user. A learning path is a sequence of learning events. When users want to perform a task, APOSDLE identifies the prerequisites. That is, it determines which learning goals are related to that task. Using the information in the User Profile, the system can determine learning needs and their order (see Section 5.3). Besides, APOSDLE proposes related topics that learners might need to study to gain understanding of the subject. We use a tree structure to present the relationships between the learning events. This presentation aims to provide context information to the learner, so that learners are able to see the relationship between the presented information and the concepts they already know and between the presented concepts themselves. An example of a tree structure with Learning Events and prerequisite knowledge is presented in Figure 6-4.

**Figure 6-4: Tree structure to present the relationships between learning events**

In Figure 6-4 the terminal nodes of the tree represent available learning events created on the basis of the current task and the learning goals to be mastered. One level up are the concepts (from the domain model, see Section 4.1) that constitute the prerequisite knowledge for being able to learn to effectively use the lower level learning events for learning purposes.
6.1.4 Remaining problems – new challenges

Although the general approach described above is clear and operational in the first prototype, there still remain problems that have not be solved yet. These will be elaborated below.

**Selecting knowledge artefacts based on criteria**

In many cases more than one instance of a certain knowledge artefact may be available. In that case rules must be available to determine which of these instances will be used in a template. Criteria have to be defined for this selection process based on a quality rating by experts or users.

Selecting knowledge artefacts based on quality means that all tagged artefacts also have to be rated on a quality scale. This is a very labour intensive process that is difficult to do automatically. An alternative procedure is to rate documents, but this means that every knowledge artefact (as a part of that document) will get the same quality rating. This could mean that a very good definition or example that is present in a document with a low quality rating, will never be used in a learning event.

An alternative would be to start with random selection and have users rate the quality of artefacts. This (average) rating will be used in the future to select artefacts. However, user ratings could be unreliable because users, who obviously lack certain knowledge, have to rate artefacts related to that knowledge. In the second prototype we will experiment with a rating procedure.

**Evaluation and feedback on engagement activities**

An important prerequisite for learning is feedback on activities. Therefore it is important to get feedback on the answers users give to questions and tasks that are specified in the engagement activity. Users can give their answers in the input fields that are available in a learning event. However, evaluating the content of these input fields by the system is impossible because of the generic architecture of the system. Beforehand it is not known what the content of the artefacts is that are displayed (only their material use type). Therefore the questions and tasks in the engagement activity are very general. This makes it impossible to formulate content related evaluation criteria that can be used by the system to assess the quality of the input.

This implies that other types of feedback mechanisms have to be implemented. One way to obtain feedback is consult co-workers who are more experienced and to discuss the content of the learning event and the engagement activity with them (see Section 6.2). Another way is to compare the results of an engagement activity with the results of others who have also done the activity and to discuss with one or more of them one’s own answers.

**Social interaction and learning**

Social interaction and collaboration are important activities for learning because during these activities people have to express and explicate ideas, discuss, give arguments and reflect on actions. Elements could be included to specify the types of collaboration activities that users can engage in and that can support different types of collaboration activities (for instance coaching, modelling).

**Identifying whether a learning goal is mastered**

To be able adjust the user profiles to the current knowledge level of the user, criteria must be identified based on which it can be assessed whether a learning goal has been attained. This is a difficult problem because assessment based on content related criteria is not possible, due to the generic nature of the environment (see also footnote 20). Therefore, alternative criteria have to be identified to assess when a learning goal is mastered and a procedure has to be in place to update the user profile.

**Keeping track of learning events that have been accessed**

The learning tool fills the learning templates with the best fitting knowledge artefacts. This means that the content of a learning event in principle will be the same every time it is generated if there are no new documents and knowledge artefacts available since the last session. To prevent this, the system
should keep track of the learning events that have been selected by the learner and should not present these events in the list the next time, or present them in a separate list with worked through learning events. This means that the user competency profile should include a learning events section that logs which events have been accessed (planned, see Section 5.3).

**Sequencing of Learning Events**

In the current prototype every learning event is self-contained and contains no reference to another learning event. For certain topics, it might be necessary to present a series of learning events instead of one learning event. Such a series could be based on instructional theories such as the “conditions for Learning” from Gagne. A series could also be based on the easy-to-complex or general-to-specific. A series of related learning events will be referred to as a learning episode (or a learning path).

**Marking relevant parts of displayed pages**

In the first prototype a page from a document is displayed in which a knowledge artefact can be found. The user has to detect the relevant part of this page. This could be facilitated by using a kind of marker to indicate to the user where the most relevant information can be found in the pages that are displayed in the learning event. In the current version of APOSDLE the user has to read the whole page to determine what the relevant part is. In the second prototype this marker idea will be implemented.

6.2 Supporting collaboration processes

The utilisation of knowledge for daily work purposes is one of the main characteristics of knowledge workers. They collect, apply and create knowledge in terms of information or data sources. In the same context the transfer of knowledge via communication and collaboration plays also an important role. APOSDLE addresses this by focusing not only on the individual work and learning situation, but also by covering collaborative ways of working with knowledge.

Collaboration support within APOSDLE focuses specifically on communication and collaboration support in learning situations. That is, it is not intended to support the entire palette of possible user to user interactions, but to investigate which interactions are especially important during a learner-expert exchange.

This section uses an activity model to illustrate the different possible collaboration/communication activities within an APOSDLE community, not all of which are supported in Prototypes 1 and 2. Depending on the type of interaction desired, relevant channel selection needs to take place. The APOSDLE approach is to first help the user choose the right person to talk to, and then to provide the user with a choice of different communication channels depending on several aspects: availability of the person to be contacted, preferences of the person to be contacted, affordances of the learning situation to be supported, etc.

Another important APOSDLE approach to communication is the contextualization of communication channels. Instead of simply offering the channels, APOSDLE aims to initialize them by conveying the context of the learner to the person being contacted (expert). This includes providing information about the current work task being executed by the learner, the relevant learning events visited by the learner, relevant documents accessed and open, etc.

In order to enable other users to learn also from an interaction between one specific learner and expert, it should be possible to make (edited) transcripts of the communication available.

---

21 Due to a change in project partners, the collaboration aspects of APOSDLE were still very much under development when writing this deliverable. A more in-depth and thorough version will be presented in a separate deliverable planned for M24.
6.2.1 Challenges

Historically, the most effective way to collaborate is in face-to-face situations when all participating individuals are able to use all verbal, mimic and gestural communication channels. This observation was also made by the APOSDLE workplace learning study (de Hoog et al., 2006) in which workers from different backgrounds were asked to provide information about their organisational work and learn situation. Replacing face-to-face communication by electronic forms may lead to a loss or misinterpretation of information because these media are not as rich. The major challenge we face is providing sufficiently rich channels for communication that enable the sharing of work contexts between communicators in order to improve learning at work.

The second challenge derives from the APOSDLE specific approach towards learning. Learning is not only considered as self directed and task driven, but also by the utilisation of interpersonal information exchange through collaboration. Here the APOSDLE workplace learning study (de Hoog et al., 2006) has worked out that 70% percent of 146 interviewed knowledge workers prefer an interpersonal information exchange. Thus, the APOSDLE learning strategy can be summarised as “learning by doing” and “learning by social interaction”. This means that it should also be possible to share learning materials and learning events to enable other workers/experts to give meaningful feedback to the learner and to stimulate reflective thought.

While most information which is used and produced in collaboration processes is bound to individuals, dealing with personal privacy will become a real challenge in APOSDLE, especially balancing between using personal data for collaboration purposes and hiding it for privacy reasons (see also Section 5.4).

6.2.2 Contextualised Collaboration in APOSDLE

The APOSDLE approach regarding the support of collaboration processes focuses on communication and collaboration as major activities in collaborative knowledge worker communities. In addition to socialising effects, the sharing of knowledge for work or learning purposes plays a crucial role for the building and performance of such collaborative communities.
Figure 6-5 shows an activity model describing the interaction between members of collaborative communities – the knowledge workers - as well as knowledge artefacts involved:

- Knowledge workers *create* knowledge artefacts by making explicit their tacit knowledge about facts, concepts and processes related to work situations, for example, using documentation techniques.
- Knowledge workers *use* knowledge artefacts for work or learning purposes by applying the obtained information.
- Knowledge workers *rate* the subjective and objective quality of knowledge artefacts with respect to their individual experiences.
- Knowledge workers *comment* on knowledge artefacts to add extra information, they discuss or reflect on their content or simply give feedback.
- Knowledge workers *tag* knowledge artefacts to classify it as a piece of information with relations to other pieces.
- Knowledge workers *discuss* about topics which are relevant to the community’s work and learning situations.
- Knowledge workers *communicate* with other knowledge workers about work or learning related topics as well as about completely unrelated topics.
- Knowledge workers *collaborate* in a certain context to share, apply and create knowledge together.
- Knowledge workers *rate* the subjective and objective competence of other knowledge workers with respect to their individual experiences.
- Knowledge workers *tag* other knowledge workers to classify their competences.

For the second APOSDELE prototype, the complete activity model will not be taken into consideration. The focus will be on supporting communication, collaboration, and commenting as well as rating activities. The next sections describe the overall approach towards contextualised collaboration in APOSDELE.

6.2.2.1 Communication and collaboration

If the demand for additional information or learning support occurs during work in the application domain, this typically takes place as part of a domain task or subtask. Then, in her role as learner, a knowledge worker has the opportunity to select from task dependent resources, learning events or learning episodes (see previous section), and in addition she can start a communication or collaboration session with other knowledge workers which are task experts or simply peers. When she wants to start such a session, it should be easy to detect which experts are available and having expertise relevant to the learning goal at hand, or which co-workers recently were engaged with the same learning goal. Furthermore, the expert or co-worker should be made aware of the task and learning goal the learner is working on. Documents or learning events should be shared (real time) to be able to engage in meaningful and task and learning relevant discussion and collaboration.

There are applications available that could support the functionality described above, like Windows Meeting Space, a collaboration program in Windows Vista which lets 2-10 users start sessions with other users for collaboration. This application supports ad hoc meetings, application sharing, file transfer, and simple messaging within a network and works primarily inside a firewall. Windows Meeting Space allows sharing of the desktop with other coworkers, distribution and collaborative editing of documents, and passing notes to other participants. It is a replacement for the older Windows NetMeeting application. However, features like microphone support and the ability to set up audio or video conferences available in NetMeeting are now removed.

In the first APOSDELE prototype, the standalone tool *ConcertChat* was used to run collaboration sessions between knowledge workers. This tool provided synchronous communication channels for
collaboration based on text chat, whiteboard and document viewing. The APOSDLE workplace learning study has shown that there is additionally a strong preference of people to collaborate face-to-face as well as via telephone or e-mail. To address these requirements the second prototype will extend the variety of different communication channels by supporting other synchronous and new asynchronous channels:

- **Audio mediated communication channels**
  While phone calls still play a significant role in intra-organisational communication, especially in larger or less technically oriented organisations, APOSDLE will provide adequate support with the second prototype. It is not yet the intention to build here only on Internet telephony. Extensions regarding this can be added in a later version. The more so if the focus is on providing context information when there is a clear separation between the APOSDLE observable world and the world outside (see also Section 6.2.2.3).

- **Text mediated communication channels**
  Text based communication takes place synchronously via instant messaging and chat communication channels or asynchronously by using e-mail. The second prototype will focus on both, because especially e-mail plays an important role in workplace oriented communication and collaboration.

Another requirement regarding collaboration comes again from the APOSDLE workplace learning study. When seeking help in work or learning situations, 63% of the 146 participants prefer working with digital documents to solve their specific problem. This leads to the addition of collaborative document authoring and discussion channels in the second APOSDLE prototype.

### 6.2.2.2 Channel Selection

The selection of a communication and collaboration channel for contacting experts or peer knowledge workers can be done manually or automatically. In the second APOSDLE prototype a combination of both will be incorporated. Based on the knowledge worker’s collaboration profile at the sending as well as receiving endpoints, the requesting knowledge worker’s choice of channels will be limited to the intersection of available and preferred channels:

- **Available communication channels** are derived from existing or provided communication tools at the knowledge worker’s workplace and technological software system.
- **Preferred communication channels** reflect the knowledge worker’s choice from existing tools that can be used to get in contact with others.

In addition, a suggestion about the optimal communication channel will be made automatically, based on the knowledge worker’s current collaboration status, preferences and situation. For example, when he is making a phone call, other knowledge workers will be suggested to contact him through an asynchronous communication channel.

### 6.2.2.3 Contextualisation and Collaboration Context Information

As described above, a key challenge for APOSDLE will become the contextualisation of communication and collaboration activities fitting the domain model and individual work or learning situation. Contextualisation can be applied on the transferred information or on the used communication and collaboration tools. The first one selects and provides the amount and depth of information (data, documents etc.) which are essential to solve a certain problem, the latter selects and adapts software which is collaboratively used for problem solving purposes. To give an example: when a knowledge worker requests help in modelling UML, he can use documents and learning events which are provided by the APOSDLE system or he can start a communication and collaboration session with an UML expert. In this case the APOSDLE system could provide in parallel to an audio communication channel, another application sharing channel that is used to share a diagram modelling application. A contextualisation based on the underlying software would mean to limit the applications functionality to UML modelling only in this example.
Contextualisation in the second APOSDLE prototype, will focus on the transfer of collaboration context information via provided channels. While in synchronous collaborations the context information can be transferred immediately from the sending to the receiving knowledge workers, in asynchronous collaborations the context information will be stored together with the communication message at least until it is fetched by the receiver.

Furthermore, a collaboration can take place separated from the APOSDLE system, for example, in traditionally linked phone calls. This needs a mechanism to bridge the gap between the digital APOSDLE world and the physical world outside. In the second APOSDLE prototype an additional context information viewing component will provide all the necessary information about communication partners, resources and learning events for the embedded communication and collaboration channels.

6.2.2.4 Collaboration Transcripts

The result of a collaboration is a history of all activities that took place (conversations, actions etc.). This history can be useful for reuse in knowledge creation processes or knowledge artefact enrichment. For this reason, it is stored as a collaboration transcript after the end of a collaboration session.

The format of collaboration transcripts depends on the native communication channel format. Thus, transcripts of text based communication channels will be stored in a text based format, audio mediated communication will be stored – as far as possible – in an audio format etc.

To combine the transcripts of more than one communication channel in parallel, there will be a meta transcript containing the included transcripts.

6.2.2.5 Collaboration Feedback

During or after the end of a collaboration session, knowledge workers will have the opportunity to give feedback through rating and commenting on its success and quality. Especially visible rating mechanisms can have here a great effect on the motivation in collaborative networks to provide help and information to others. APOSDLE will make use of this effect to add a more socially driven approach to collaboration in work or learning situations.

In the second APOSDLE prototype, there will be functionality to rate the quality and success of a collaboration session and thus, implicitly, the expert’s quality. Furthermore, knowledge workers will have the possibility to comment on peers or experts to give more in-depth feedback. This information can become part of publicly available information about knowledge workers as long as organisational, personal or legal reasons do not forbid it.

6.2.2.6 Privacy Issues

Any time when personal data is recorded or even stored this has implications at least from a legal perspective (see also the discussion about privacy in Section 5.4). The European law clearly defines the limitations of personal data processing:

- persons from which data is processed have to be informed about its usage and processing reasons as well as they have to agree on it,
- persons must have the possibility to view, change or delete their recorded data at any time and
- it must be secured that personal data is protected against unauthorized usage.

This also would have strong effects on the use of contextualised collaboration in APOSDLE, because any collaboration profile and transcript, as well as ratings and comments, are personal data and fall under the mentioned limitations. In addition personal privacy is also an issue.

For this reason, the collaboration part will follow the overall privacy policy of APOSDLE. The knowledge worker should decide which data he is willing to publish and which should stay private.
6.2.3 APOSDLE collaboration framework

The technological concept for supporting collaboration processes is based on an adaptable software framework which can be extended any time with new communication and collaboration channels or collaborative network functionality. The framework contains a core component which is responsible for the following tasks:

- Providing context and knowledge dependent communication, collaboration and authoring services by plugging in corresponding software tools,
- Performing, documenting and archiving communication, collaboration and authoring/re-authoring activities,
- Producing and ensuring awareness within the communication, collaboration and network processes.

Furthermore, the framework aims at adapting or enhancing existing tools to provide communication channels (email, chat, telephone), collaboration services (whiteboard), collaborative network services (wiki, rating, commenting) as well as authoring/re-authoring services (web authoring). For the second APOSDLE prototype, the following tools will be used for communication and collaboration:

- MediaWiki
- Standard email clients (Thunderbird, Outlook, Outlook Express)
- ConcertChat and TextChat
- Telephone
7 Knowledge Artefact Lifecycle

This chapter describes the lifecycle of knowledge artefacts. First, a document enters the APOSDLE system as a repository object (for example, during system instantiation, during nightly updates, after communication events). Next, the entire document or parts thereof are turned into knowledge artefacts by manually or automatically attaching two types of metadata: domain concept(s) and material use(s). Finally, the knowledge artefacts are related to each other and retrieved via an associative network. First ideas on how user feedback mechanisms on the different steps can be utilized are included. The individual steps are described below.

7.1 What is an APOSDLE knowledge artefact?

In APOSDLE a knowledge artefact is defined as a document, or part thereof, together with two types of metadata: the learning domain concept (see Section 4.1) addressed/described within the document (piece) and the material use type (see Section 6.1.2) of the document (piece). Optionally a knowledge artefact can also have a free text comment associated with it.

7.2 Access to knowledge artefacts

Explicit knowledge of a company, incorporated in documents (reports, articles, manuals, video and so forth) is often stored in various back-end systems (repositories) in this company. As APOSDLE aims at taking into account the entire intellectual capital of a company and make it available to the knowledge workers, this explicit knowledge has to be accessed somehow. Documents retrieved will further serve as a basis for creating knowledge artefacts, which themselves will be used within APOSDLE for learning or collaboration.

7.2.1 Challenges

Concerning the access to back-end systems some challenges can be identified that have to do with unified access and privacy.

IT-Systems of modern companies are based on a highly networked infrastructure using different operating systems with various interconnection and communication protocols. Due to this variety of back-end systems, a first major challenge is to provide a single point of access for all modules APOSDLE consists of. Whenever an APOSDLE component wants to retrieve documents, it should not matter where this documents originally resides. For this purpose, a general interface may be used which provides homogeneous access to all documents, independent of the underlying infrastructure the documents are stored in.

Especially in companies with many employees, privacy issues play an important role. There are various access restrictions based on user or group rights. For prototype 1 this was omitted but for prototype 2, another challenge in APOSDLE is that privacy issues concerning document access rights need to be taken into account. Not all users of the APOSDLE System are allowed to access all documents stored in the back-end systems (see also Section 5.4 for possible solutions).

7.2.2 Theoretical Background

The theoretical background of our work is based on technologies like GRID computing. Hyatt and Vrabik (2004) define Grid computing as distributed computing over a network of heterogeneous resources, enabled by open standards. The special “Information Grid” tries to provide secure access to any information source (heterogeneous files, databases, and storage systems) regardless of where
it resides. To achieve this, so-called virtualization of data across various back-end systems is used. With virtualization, the data appear to come from one source even though the data is stored in a distributed way.

APOSDELE’s implementation of a homogeneous access to various back-end systems was inspired by the Java Content Repository (JCR). The idea behind the JCR is that with the growing popularity of content management applications, the need for a common, standardized API for content repositories has become apparent. JCR tries to provide such an API by means of having an interface that can be layered on top of a wide variety of existing content repositories. The JCR architecture consists of a so-called repository model holding different workspaces. Each workspace consists of nodes and properties. The homogeneous access approach used in APOSDELE uses somewhat the same architecture, although it is not exclusively related to content management systems. There is one interface used by other APOSDELE components (the Data Object Repository) which connects to various back-end systems. The back-end systems store the documents.

### 7.2.3 APOSDELE Approach

To guarantee homogeneous access to all documents stored in different repositories, a so-called Data Object Repository is used. The Data Object Repository provides a general interface and therefore a single point of access for all other modules and tools within the APOSDELE System (see Figure 7-1). Documents can be retrieved on the basis of a unique identifier created for each document. Within the Data Object Repository and the entire APOSDELE System the document itself is represented by a so-called Repository Object (RO).

![Figure 7-1: Data Object Repository within APOSDELE Platform](image)

Repository Objects can be understood as a document representation enriched with automatically obtained information in terms of metadata (like mime type or file size). After creating a Repository Object,
Object based on a document, this RO is stored and made available to the APOSDLE System. The following issues concerning a Repository Object must be addressed:

- Each Repository Object is identified by a unique identifier within the APOSDLE System. One possible solution for this is to use a Global Unique Identifier (GUID) as proposed in RFC 4122 (RFC4122, 2005).
- Based on this unique identifier the corresponding document may be retrieved. The content is further used, for example, for indexing and classification by the APOSDLE’s Knowledge Discovery and Extraction Service.
- Furthermore, a Repository Object may provide some system-managed, read-only additional information (metadata) based on some standardization of digital objects like the Dublin Core Metadata Element Set (Dublin Core). Metadata may consist of Document-Title, the GUID, mime type, file size and so forth.
- Concerning privacy, Repository Objects underlie an access mechanism, which, in our case, is primarily based on the permissions of the corresponding document within the appropriate repository integrated into the APOSDLE platform. Therefore, the access permissions have to be resolved from the repository systems, for example, by utilizing mapping techniques.

We choose this approach of using a Repository Object because the Repository Object represents a general interface for other APOSDLE components to deal with documents and corresponding metadata within the APOSDLE System.

The Data Object Repository, as one module of the APOSDLE System, handles both the Repository Objects and the connected repositories. Repository Objects are stored in a database for further retrieval by other modules. A so-called Repository Manager deals with the repositories used in the APOSDLE System. Repositories may be located in or outside the APOSDLE Platform. Each repository is defined by having one out of several repository types and additional information about how to access the appropriate back-end system. A repository type represents a local or remote File system, a connection via WebDAV or Samba, an Email Server etc. Technologically, the Product Factory design pattern (Metsker & Wake, 2006) is applied within the Repository Manager. Using the Repository Manager, repositories may be managed (activated/deactivated) on-the-fly.

One issue concerning the Data Object Repository is to detect changes in the connected back-end systems (updated or deleted files), so that the APOSDLE’s Knowledge Discovery and Extraction Service can re-classify the data and the APOSDLE System is able to present the most accurate results to the knowledge workers. Determining the changes in the back-end systems is done on a periodical basis.

Concerning repository types, the Data Object Repository is build up quite modular. Access to the different repository types is implemented using 3rd Party libraries and APIs using the same overall interface.

We choose this approach because it has to be flexible in adding additional repository types: each Company’s IT-Infrastrucure APOSDLE will be used in, will be (very) different. Using a database storage for the Repository Objects is necessary to determine changes. Detecting those changes is performed on a periodical basis, because hardly any back-end system provides a proper and non proprietary notification mechanism.

**7.2.4 Where else has something like this been used and proven to work**

A similar approach of gathering data from back-end systems has been shown in (Aperture 2007). Aperture is a Java framework including an open source library for extracting full-text content and metadata from various information systems (for example, file systems, web sites, mail boxes) and the file formats (for example, documents, images) present in these systems. The Aperture framework is created in cooperation between the German DFKI institute ([http://www.dfki.de](http://www.dfki.de)) and the Dutch software
firm Aduna (http://aduna.biz) and is currently in an Alpha-stage. APOSDLE’s Data Object Repository will partly benefit from Apertures functionality. Compared to APOSDLE, the framework uses a crawler-based architecture and is not intended to provide access to single documents. Furthermore it lacks privacy management.

7.2.5 New challenges and ideas to solve them

During the implementation of Prototype 1 and the planning phase of Prototype 2 several new challenges emerged.

A key idea behind the APOSDLE System is that documents retrieved using the Data Object Repository are presented to the knowledge workers, based on their actual work context. Knowledge workers are then able to annotate those documents (respectively parts of them), for example, with domain concepts. This is the process of creating knowledge artefacts on the basis of the documents (see also the next Section). Due to the huge amount of different file formats found in the back-end systems, it is almost impossible to provide such an annotation facility for all those file formats. So the idea to solve this, presenting a new challenge, is to convert the different files into a format that can be easily displayed and annotated. Plans for Prototype 2 of APOSDLE are to use PDF as a target format. For the Data Object Repository, as underlying infrastructure, this means using different concepts in delivering the converted documents and dealing with access rights.

As strategies to prevent unauthorized access to documents have to be taken into account for Prototype 2 new challenges in dealing with the APOSDLE Application Partner’s existing Access Control architecture arise. In the Application Partner’s IT-Infrastructures we have to cope with highly secured intranets containing more or less confidential information. For this reason it will be quite difficult to access this information for a prototype application like the APOSDLE System, which is not fully integrated into the Partner’s IT-Security concepts. For this purpose we will on one hand deal only with non confidential information extracted from the productive systems. On the other hand we try to find solutions to gain access to the back-end systems without compromising the security architecture. This could for example be done by using some sort of single-sign-on mechanisms as used in various network architectures (see, for example, Microsoft 2007).

7.3 Creation of knowledge artefacts

One essential step in the APOSLE knowledge artefact lifecycle is how knowledge artefacts are created. One approach in creating knowledge artefacts is to do this in a collaborative fashion, where users create knowledge artefacts by annotating documents with a domain concept and/or a material use as described below. Due to the huge amount of knowledge items targeted by APOSDLE, one single user or a small user group will not be capable to annotate all knowledge artefacts. Therefore, tools must support a collaborative annotation, so that users can share their knowledge about knowledge artefacts with each other.

However, manual annotation is a labour intensive task and we do not necessarily expect manual annotations to provide training data covering all necessary concepts for classification. To overcome this natural limitation and to bootstrap an annotation free repository, APOSDLE makes use of linguistic and machine learning approaches to automatically assign domain concepts and material uses to knowledge artefacts.

7.3.1 Challenges

As defined, knowledge artefacts are documents or parts thereof enriched with metadata. Following the APOSDLE concept of the 3spaces concerning knowledge worker, learner and expert they are mainly needed for creating learning events and displayed as usable resources for the knowledge worker (see Section 6.1). So the basic challenge is to extract parts of data objects, enrich them with specific
metadata and present them to the knowledge worker. This includes providing different possibilities for adding metadata to integrated data objects.

7.3.2 APOSDLE Approach

In APOSDLE knowledge artefacts are created in two ways. On the one hand knowledge artefacts can be created manually by annotating documents or media files. On the other hand they can be created automatically, based on learning from given manually annotated test cases.

7.3.2.1 Manual creation of Knowledge Artefacts

Among others, manually created artefacts serve as bases for further automatic creation of knowledge artefacts. In APOSDLE the manual creation of artefacts is supported by the usage of the so called Annotation Tool. This tool not only helps the user to enrich parts of integrated documents or media files with specific metadata (material-use, semantic concept, free notes), but it is also used for presenting knowledge artefacts and their assigned annotation. Therefore, from a user’s point of view the annotation tool is more a support tool for her daily work than a tool for annotating documents. This approach follows successful Web 2.0 approaches which have shown that annotation has to be embedded in the knowledge task rather than being a separate tasks. Similar findings were reported by (Cutrell, Robbins, Dumais, & Sarin 2006). Their experiments, conducted at Microsoft Research, revealed, that while users are willing to annotate documents, the annotation process must be included in tools they use for their work, as done in the APOSDLE approach.

Concerning documents firstly the Annotation Tool can deal with Portable Document Format (PDF). Before a user can annotate documents they have to be converted into PDF. The conversion is performed by the Data Object Repository whenever a new document is integrated into the APOSDLE system (see section 7.2.5). Secondly, videos encoded in a common format can be used.

The first step in manually annotating documents (files) is to select one specific document by using the Repository Object Browsing Tool (ROBT). The ROBT is a graphical tool, which lists all available data objects in the APOSDLE System, independent of the backend system where the data objects are actually stored.

By selecting the data object in the graphical user interface (GUI) of the ROBT the Annotation Tool is started. In case of a text document the Annotation Tool retrieves the content of the converted PDF via a Web Service Request to the APOSDLE Platform. In case of a media file the media file content is retrieved.
The annotation process itself is done as shown in Figure 7-2, and described briefly below. A part of the document is selected (grey area in left hand pane) and the appropriate metadata is chosen from a list of available metadata (domain concepts: upper right hand pane; material use type: central right hand pane). The position of the marked text or video fragment in combination with the material use and a concept is stored and a new knowledge artefact is created. This step can be repeated for creating further artefacts. The list of available metadata can be configured for different environments. When the annotation process is finished, the newly created knowledge artefacts (containing, for example, assigned concept, position in text, etc.) are stored in the structure repository and listed in the Annotation Tool. These artefacts can be changed or deleted at any time.

7.3.2.2 Automatic creation of knowledge artefacts

Automatic creation of knowledge artefacts distinguishes between the annotation of material use and the annotation of domain concepts to document parts (see also Figure 7-2). Both approaches utilize supervised learning techniques which require learning a model from previously annotated documents (for example, training data). The model is afterwards used for annotating new, previously unseen, artefacts.

From an architectural point of view the functionality is embedded in the APOSDLE architecture via the Annotation Service. The service interacts with the structure and document repository to load the training data and to store the newly created annotations. Functions are triggered via the coordinator and results are obtained from the Annotation Tool by directly accessing the structure repository. Due to its loose coupling with other components, the Annotation Service may be of further use in future scenarios, like annotating web pages on the fly.

The AnnotationService has basically two modes of operation:

- training of a classification model for the annotation of documents (material use, domain concepts) based on manual annotations from knowledge workers using the Annotation Tool.
Both activities are triggered by the “Coordinator” from within the APOSDLE framework. Based on the mode of operation the AnnotationService performs the following, simplified workflow:

- **Training**: Get all manually annotated knowledge artefacts from the Structure Repository and train a classification model for each view (material use, domain concept).
- **Classification**: Iterate over all unprocessed documents from the Document Repository. Find and classify knowledge artefacts within each document and store the results in the Structure Repository as “automatic annotation”.

For both, the Common Access Tool is needed for a normalized view on the documents (PDF), so that each component (Annotation Tool, AnnotationService) operates on the same input. In the learning phase, the annotation service fetches a document from the document repository together with annotated concepts and material uses.

From an algorithmic perspective, the document is pre-processed using standard text mining approaches like tokenisation, sentence splitting and part of speech tagging. Since parts of documents should be annotated, the document is split into parts following two strategies. The top down approach analyses the document structure (for example, headings, number of line breaks etc.) and tries to identify meaningful blocks. The blocks are annotated afterwards. While exploiting a documents structure is preferable, structural elements are not recognizable for some document formats. Therefore a bottom up strategy is applied in addition. This approach starts at the sentence level and merges succeeding sentences, which are annotated with the same material use or domain concepts. Future activities also plan to incorporate more sophisticated counting strategies like majority votes or weighted majority votes.

The annotation of material use(s) in a document will be performed by utilizing inherent semantical and structural information. For example, a material use of the type “Question” will consist mainly of sentences which close with a question mark. Such information will be used to train a statistical model based on the maximum entropy principle, which is currently state-of-the-art for statistical language processing. Maximum Entropy modelling has been proven to be a valuable tool for a broad range of language processing tasks (see (Ratnaparkhi 1998)), and therefore has been chosen as the approach for knowledge artefact labelling.

Assignment of domain concepts can be seen as classical text classification problem of assigning documents to a set of predefined classes (Sebastiani 2002). From the accurate and fastest algorithms, Support Vector Machines (SVM’s) and k-Nearest-Neighbour algorithms seem to perform well in most test cases. Since k-Nearest-Neighbour algorithms do not require excessive computing time and can be update in an iterative manner with lesser effort than Support Vector Machines, we choose to use k-Nearest-Neighbour algorithms for the assignment of domain concepts. Especially in the case of dynamic repositories, iterative update is of great importance. Since the given domain structure is hierarchical in nature, the classifier has been modified to consider hierarchical relationships to increase accuracy (Granitzer & Auer, 2005).

7.3.2.3 **Displaying a knowledge artefact**

Of special importance is the visual representation of annotations, which is not only used during the annotation process itself, but also for displaying relevant document parts with respect to a users information need. Providing such a visual overview has been demonstrated to greatly enhance the perception of relevant information while reducing the cognitive burden and information overload induced by only textual content (Kienreich et. al. 2005).

In general the visualisation is based on the theme river paradigm (Havre, Hetzler, Whitney & Nowell 2002): Given a set of annotations, each occurrence of an annotation is visualised in a side bar (see Figure 7-3, central pane; see also Figure 7-2) where the vertical axis denotes the position of the occurrence of the annotation in the document and the horizontal axis denotes it’s confidence or relevance. Thereby, the beginning of the vertical axis denotes the beginning of the document and the
end of the vertical axis the end of the document respectively. A marker line is used to show the current position in the document.

The relevance of an annotation thereby comes from two possible sources: one source is the automatic annotation. In this case the relevance shows the confidence of the automatic annotation tool in assigning an annotation to a specific document position. A second source for the confidence is the query a document was opened from. In this case, the relevance shows the importance of an annotation with respect to the query a document has been opened from. While the latter source is of higher importance for the knowledge worker, since it immediately shows relevant document parts, the first sources is helpful in validating the automatic annotation process. In both cases, it is expected that navigation within large documents will be greatly enhanced by using the adapted theme river visualisation.

![Figure 7-3: Displaying existing knowledge artefacts of a document](image)

7.3.3 Why did we choose this approach?

In APOSDLE we opted for the integration of manual annotation with automatic annotation because the difficulty is that manual annotations are in general better than automated ones, but also more expensive to obtain. Our approach therefore is to motivate users to manually seed-annotate selected documents and do the rest automatically. Over time, both more manually created knowledge artefacts will exist and the classifier will be better trained.

Due to the huge amount of available text document formats it is nearly impossible to develop an Annotation Tool, which supports every single format (concerning, for example, text selection). For this purpose the conversion to PDF is performed, so that the Annotation Tool has only to deal with PDF.

We use a video player capable of viewing common video formats and that has the ability to be set to a predefined position to jump directly to the beginning of a knowledge artefact.
7.3.4 New challenges and ideas to solve them

7.3.4.1 Defining metadata for annotation

Concerning environments of the different application partners, one challenge is to define the set of metadata, which fits best to the partner's workflow. This is done by analyzing the core processes the corresponding application partner deals with.

7.3.4.2 Automatic creation of knowledge artefacts

One obvious problem in using machine learning methods for automatic annotation is the number of training examples required for learning the accurate classification model. Especially if the documents differ in style and content, a large number of training examples is required. The following solutions have been foreseen to tackle this challenge:

1. **Feature selection and development of different heuristics for annotating material uses.**
   Since there are only a few classes of material use, in contrast to the larger number of domain concepts, different models are learned for each material use based on different, handcrafted heuristics. Furthermore, each material use is based on different, manually selected feature selection strategies, reflecting special properties of a material use. By manually tuning the material use models in this way, human a-priori knowledge can be incorporated thereby reducing the amount of necessary training examples. A combination of learning and rule based approaches would be a natural extension to this.

2. **Bootstrapping training examples using unsupervised approaches**
   For annotating domain concepts it is not possible to tune every domain model due to the number of concepts and changes from domain to domain. Therefore, we rely on bootstrapping training examples using statistical patterns and unsupervised learning approaches. In particular, all documents are analysed in an unsupervised manner by extracting associations between terms of knowledge artefact. Those associations are used to extend the classification hypothesis and to consider also terms not contained in the training examples. Another approach for annotating domain concepts would be the use of transductive learning techniques like, for example, transductive support vector machines (Zien, Brefeld, & Scheffer 2007), which is planned if performance estimates of our current approach fall below expectations.

3. **Accuracy expectations**
   One important aspect in applying machine learning techniques is the achieved accuracy by automatic means. Before applying such techniques, only rough guesses can be made on how a system will perform and how the algorithm meets the expectation of the user. Depending on the results of our evaluations, means for increasing the quality of feedback may be foreseen. So for example users not only assign annotations to a paragraph or not but may also state the important reason(s) why an annotation is assigned or not. While this process is more labour intensive, it may yield more accurate results in future.

7.3.4.3 Usability of Annotation Tool

For the process of manually annotating documents one challenge is to present the user a user-friendly tool. This includes the presentation of the metadata, the document and the artefacts. Due to the fact that the Annotation Tool is used for different tasks, like creating knowledge artefacts, displaying knowledge artefacts, changing – deleting knowledge artefacts, browsing all knowledge artefacts of a specific document and all of these things as well for PDF as for videos, the challenge is to provide a clear functionality and combine the different functionalities in one tool.
7.4 Retrieval of knowledge artefacts

7.4.1 Challenges

The environment the knowledge worker operates in, can be represented by formal domain models which conceptualize those entities of the worker’s domain, which are relevant for work-integrated learning. For example a model for the domain of requirements engineering would define which methods exist for the elicitation of engineering and how they are related. Formal models define the entities of a domain and the semantic dependencies between them by using a formalism, for example, a special language. APOSDEL uses a set of models to formalize the context of the knowledge worker at a company where APOSDEL is deployed (see Chapter 4). This raises challenge 1: the retrieval of resources for learning based on entities within the models.

To use the information present in the models for the retrieval of learning material available in the document base of a company, the models and the document base have to be related by some means. Per se those two tiers are not connected, as the domain models are created by experts for a special purpose, while the resources in the organisational memory are created by the workers in the company during their daily work. This leads to challenge 2: the relation between the models and the resources present in the document base.

To interrelate the two tiers and thus make the documents usable for retrieval and the creation of learning material from them, the entities present in the models are used to annotate the resources in the document base of the company. By this process, concepts stemming from the models become semantic metadata to the resources present in the document base.

Two options are possible for assigning semantic metadata to the available resource:

1. Manual assignment of semantic metadata – which is a time consuming task
2. Automatic assignment of semantic metadata – which does not equal the precision of metadata assigned by human experts.

The difficulties associated with assigning semantic metadata can easily lead to a sparse annotation with semantic metadata of resources in the document base: only a few concepts from the models are used for annotation and only a few resources are annotated.

Sparse annotation of resources with semantic metadata, however, burdens the search process for resources based on this metadata: A search based on a set of concepts will not be able to identify resource relevant for a query if the resources in the document base do not have semantic metadata assigned. This brings us to challenge 3: retrieval of resources despite of the sparse annotation of resources with semantic metadata.

7.4.2 APOSDEL approach

In this section we will address the first and third challenges presented above, the second one was taken up in Section 7.3.

7.4.2.1 Challenge 1: retrieval based on semantic metadata

APOSDEL uses an associative network as the foundation for the context-based retrieval of knowledge artefacts. An essential element of the network are the semantic annotations of knowledge artefacts. A similar problem is present in establishing the semantic web: many resources already exist in the current web, but most of them are not annotated semantically. Together with the large number of resources already present on the web (or in a company) this is one of the major obstacles in establishing a broad use of semantic technologies. Therefore this research project will precisely follow current and future developments in fields such as ontology learning (Buitelaar et al., 2005) and the (semi-) automatic creation of semantic metadata (Handschuh & Staab, 2003).

23 This challenge is addressed with the annotation facility of the Domain Modeling Tool (see Section 4.1).
created with the Domain Modelling Tool (see Section 4.1), which are modelled as edges in the network, connecting nodes that represent concepts and nodes that stand for knowledge artefacts. The network is queried by the User Profile Service (see Section 5.3) to retrieve knowledge artefacts corresponding to the current context of the user. These knowledge artefacts are then wrapped into learning events and presented to the learner (see Section 6.1) or are directly displayed as resources in the sidebar.

Queries in the APOSDEL system currently are formed by sets of concepts and the retrieval process relies on the semantic metadata assigned to resources in the document base. For processing the information present in the Associative Network a technique called spreading activation is employed. Initially a set of nodes in the network, representing concepts from the search query are activated. From this set of nodes, activation spreads over the edges in the network to the nodes in their neighbourhood. Those nodes with the highest level of activation are seen to be the most similar to the set of nodes activated initially and are returned as retrieval results. At the moment only nodes representing knowledge artefacts are taken into account as result to a search.

We see APOSDEL as a retrieval approach for the Semantic Desktop, as we use technology originally developed for the Semantic Web to build a desktop application to support the knowledge worker. In our approach, we aim at combining database-like queries to a knowledge representation as found in semantic web technology with classical information retrieval approaches, that is, the statistical analysis of document content.

7.4.2.2 Challenge 3: Sparse annotation of resource with semantic metadata

The difficulties associated with assigning semantic metadata can lead to a sparse annotation with semantic metadata of resources in the organizational document base. This sparse annotation hinders search for resources based on semantic metadata as a search based on concepts might not be able to identify resource relevant for the query due to a lack of documents annotated with semantic metadata.

The APOSDEL approach to information retrieval addresses this by employing a technique called Associative Retrieval in order to find material not directly annotated with certain concepts but indirectly associated with these concepts. We retrieve material that is not directly relevant to a query but – by the association with relevant material – has the potential of being relevant.

We perform our associative retrieval approach based on the semantic metadata available from the documents and based on the content of the documents to increase the amount of relevant material which can be provided. For this purpose, we exploit associations between concepts in the domain model and associations between documents in the document base of the company. Associations are created by means of semantic similarity (provided by the Semantic Service for the elements of the domain model) and textual similarity (provided by the Classification Service for the documents in the document base) and are modelled as edges in the Associative Network (see Figure 7-4).
7.4.3 Why did we choose this approach?

The APOSDLE solution relies on semantic metadata assigned to resources within the backend systems of the companies where APOSDLE is deployed. This semantic metadata is used to identify resources suitable for the knowledge worker to learn about a certain topic. At present assigning of semantic metadata is either done manually or semi-automatically, with both approaches introducing difficulties that can lead to a situation in which not the entire document base is annotated with semantic metadata.

This sparse annotation of resources with semantic metadata makes search for resources based on this metadata difficult. Search based on concepts will not be able to identify resources relevant for a query if they lack semantic metadata. We make use of techniques from associative retrieval to find resources that are not a direct result to a query but are – by the association with relevant material – promising of being relevant.

The use of the associative network as underlying structure for the retrieval process allows for both the search for resources based on semantic metadata (addressing challenge 1) and at the same time allows for addressing the case of a low coverage of the resource in the document base with semantic metadata, by applying methods from associative retrieval.

As associative retrieval is quite important, we will briefly provide, in the next section, some theoretical background for this approach, also including references to data retrieval relying on semantic metadata.

7.4.4 Theoretical Background

7.4.4.1 Associative Retrieval

Crestani (1997) characterizes Associative Retrieval as a form of information retrieval which tries to find relevant information by retrieving information that is by some means associated with information that is already known to be relevant. Information items which are associated can be documents, parts of documents, extracted terms, concepts, etc. The idea of associative retrieval goes back to the 1960s,
when investigations (Salton, 1963; Salton, 1968) in the field of information retrieval tried to increase retrieval performance using associations between documents or index terms, which were determined in advance. Crestani (1997) presents a detailed survey of existing approaches to associative retrieval.

7.4.4.2 Associative Networks

In associative retrieval, association of information is frequently modelled as a graph, which is referred to as an Associative Network (Crestani, 1997). Nodes in this network represent information items such as documents, terms or concepts, edges represent associations between information items. Edges can be weighed and / or labelled, expressing the degree (and type) of association between two information items.

7.4.4.3 Spreading Activation

Spreading activation originates from cognitive psychology (see Anderson, 1983)) where it serves as mechanism for explaining how knowledge is represented and processed in the human brain. The human mind is modelled as network of nodes, which represent concepts and are connected by edges. Starting from a set of initially activated nodes in the net, the activation spreads over the network (Sharifian & Samani, 1997). During search, activation flows from a set of initially activated information items over the edges to their neighbours. The information items with the highest level of activation are seen to be the most similar to the set of nodes activated initially.

Spreading activation found its way into applications in both neural and semantic networks for information retrieval (Crestani, 1997). It is comparable to other retrieval techniques regarding its performance (Mandl, 2001).

A detailed introduction to spreading activation in information retrieval can be found in (Crestani, 1997). A description of our studies on the topic can be found in (Scheir & Lindstaedt, 2006).

7.4.4.4 Retrieval based on semantic metadata

While retrieval based on metadata is a well known paradigm in information retrieval research, semantic metadata introduces new aspects into the retrieval process. In earlier metadata schemas, such as taxonomies or gazetteers, elements from the knowledge organization scheme have been related by a distinct set of relation types. In the case of semantic metadata the elements in a knowledge representation are related by arbitrary semantic relations, which can be taken into account for retrieval purposes (Lux et al., 2006).

7.4.4.5 Semantic Desktop

The Semantic Web (Berners-Lee et al., 2001) movement led in recent years to the development of new, standardized forms of knowledge representation and technologies for coping with them, such as ontology editors, triple stores or query languages. The Semantic Desktop paradigm (Sauermann et al., 2005; Decker & Frank, 2004) stems from the Semantic Web movement and aims at applying technologies developed for the Semantic Web to desktop computing to finally provide for a closer integration between (semantic) web and (semantic) desktop.

7.4.5 Where else has something like this been used and proven to work

7.4.5.1 Information retrieval in the Semantic Web and on the Semantic Desktop

A growing amount of proposed models and implemented systems exist, which make use of semantic metadata for information retrieval. The work of Helfin and Hendler (2000), Shah et al. (2002) or Guha et al. (2003) can be viewed as pioneering work in this domain. Kiryakov et al. (2004) and Castells et al. (2007) present logical consequences of this early work with fine tuned, evolved systems.

Applications of semantic metadata lead from indexing it together with textual data (Shah et al., 2002; Chirita et al., 2006) over extensions of the vector space model (Kiryakov et al., 2004; Castells et al., 2007) to modelling documents as parts of knowledge bases (Zhang et al., 2005). Beagle++ (Chirita et
al., 2006) is an example of a search engine for the Semantic Desktop and SEAL is an approach ranking search results in semantic portals (Stojanovic et al., 2001; Stojanovic et al., 2003). A detailed overview can be found in Esmaili and Abolhassani (2006) and Scheir et al. (2007).

7.4.5.2 Network models, spreading activation and associative retrieval

A detailed survey of classical approaches to associative retrieval and spreading activation can be found in Crestani (1997). More recently, (Mandl, 2001) concludes from the good results of spreading activation based text retrieval systems PIRCS (Kwok et al., 1996) and Mercure (Boughanem et al., 1999) at the TREC24 that Spreading Activation based retrieval models are comparable to other information retrieval approaches in their performance for the retrieval of text documents.

Besides the aforementioned systems that operate on textual data, also approaches exist that make use of spreading activation to search in a knowledge based system. Ontocopi (Alani et al., 2003) identifies communities of practice in an ontology using spreading activation based clustering. Rocha et al. (2004) present a hybrid approach for searching the (semantic) web, that combines keyword based search and spreading activation search in an ontology for search on websites. Berger et al. (2004) present a tourist information system whose underlying knowledge base is searched using spreading activation. Finally, Huang et al. (2004) address the scarcity problem25 in recommender systems using a network-based associative retrieval approach.

7.4.6 New challenges and ideas to solve them

7.4.6.1 User feedback

At present, the users of the APOSDLE system have no option of influencing the results of the retrieval process and to bring their knowledge about the relevance of the retrieved material into the system. This is the case even if they identify results which are very bad – and thus should not be included in the results set – or very good - and thus should have a higher rank in the list of results. We plan to employ feedback mechanisms where users can state to what degree they have been satisfied by the presented results. This on one hand, allows for updating the network structure used by the retrieval component to provide better results in the future and, on the other, should give the users a higher degree of satisfaction, as they have the option to actively fine-tune the system’s retrieval performance.

Technically we plan using back-propagation for introducing the relevance judgements of the users into the network structure. Back-propagation is identical to the process of spreading activation but is performed in reverse: Those paths that are traversed from a result node to the original source node(s) of a query will receiver higher (for positive feedback) or lower (for negative feedback) weights.

7.4.6.2 Combination of concept and term based search

At present, retrieval of resources is done using a set of concepts as a query. This means that queries like “give me all resources that deal with modelling” can be resolved, with “modelling” being a concept from the domain model. What is not possible at the moment is that a user starts a search for resources that deal with topics that are not modelled in the domain model or that he or she narrows a search based on concepts from the domain model by using terms (words). To increase the options of the knowledge worker in the context of search and further enhance the possibilities provided by retrieval in APOSDLE, we plan to add the possibility to query the network structure using concepts and terms (words) at the same time.

---

24 Text REtrieval Conference

25 In the field recommender systems sparse associations between users of the systems and recommended items, leads to a similar problem as discussed by challenge 3: items can not be recommended adequately if there is not a decent set of associations between users and items.
On a conceptual level, this will be realized by creating a node in the associative network for every term that is present in the resources in the document base of the company and by taking these term nodes into account for the search process.
8 References


Engelmore, R. & Morgan, T. (1988). *Blackboard Systems*. Addison-Wesley,


Granitzer M., Auer P. Experiments with Hierarchical Text Classification, Proceedings of 9th IASTED International Conference on Artifical Intelligence, ACTA Press, Benidorm, Spain, 2005, IASTED


Merrill, M. D. (2002). First principles of instruction. Educational technology research and development, 50, 43-59.


9 Glossary

**Associative network:** An associative network (Crestani, 1997) is a generic network of information items represented by nodes. Relations between information items are expressed by edges interconnecting the nodes. Weights on the edges can be used to represent the strength of the relation between the information items.

**Associative retrieval:** Associative Retrieval (Crestani, 1997) is a form of information retrieval which tries to find relevant information by retrieving information that is by some means associated with information that is already known to be relevant.

**Collaboration Profile:** A collaboration profile is part of the user profile and contains information on personal preferences and habits regarding collaboration.

**Collaborative Community:** A collaborative community is a social group of knowledge workers within the context of a larger organisational entity which share knowledge in work or learning situations.

**Communication Channel:** A communication channel is a medium which transfers the communication message from sending to receiving communication partners. Communication channels can be divided in synchronous and asynchronous channels. Synchronous communication channels connect two or more partners at the same time. The communication message will be delivered directly with a short medium based delay. Asynchronous communication channels do not connect communication partners immediately. The communication message will be saved by the medium until it is fetched by the receiver.

**Data Object Repository:** ought to be part of APOSDLE’s underlying knowledge infrastructure and deals with managing Repositories and Repository Objects.

**Domain model:** A formal model of one domain of knowledge which is relevant to the knowledge worker. “One” domain of knowledge stresses the fact that the knowledge worker might move from one domain of knowledge to another. “Formal model” refers to the fact that the model is machine-readable. Human-readable definitions of model entities may be provided for convenience. The domain model consists of definitions of basic domain concepts, relations between them and restrictions on both concepts and relations (axioms)

**Knowledge Artefact:** a digital entity which has been created in a knowledge-based activity. Examples for Knowledge Artefacts is a (part of a) requirements engineering document or a scene in a video. Every Knowledge Artefact in APOSDLE is addressable via a unique identifier. Knowledge Artefacts can be described by a set of concepts from a formal model and related to other Knowledge Artefacts. They can be parts of documents, as long as they are addressable via a unique identifier.

**Knowledge State:** A knowledge state is given by a set of tasks and the associated set of knowledge and skills needed to perform these tasks. A learner is said to be in one knowledge state if he/she has been sufficiently engaged with this set of tasks and the associated skills and knowledge either through performing the tasks or through learning. This state is then called the competence of the learner. Because of the prerequisite relation on learning goals, not all knowledge states are permissible.

**Knowledge worker:** an employee of an organisation whose essential operational and value creating tasks consist in the production and distribution of knowledge

**Learning episode:** A series of related learning events.

**Learning event:** A learning template related to a learning goal that is filled with knowledge artefacts and references to co-workers and/or experts.
Learning Goal Interpretation: The learning goal interpretation of a task is the set of knowledge and skills which are required at the minimum for performing the task.

Learning Goal Type: The type of knowledge or skill which the learner wants to acquire. This is expressed as a cognitive activity which a learner can perform after acquiring the knowledge or skill (i.e. remember, understand, apply, create) with regard to a certain element of the domain ontology.

Learning Goal: One skill or element of knowledge which a learner wants to acquire at a certain point in time. A learning goal is expressed as a tuple of a domain model element and a learning goal type.

Learning History: The collection of all learning goals which the learner has been engaged with in the past. This may have happened either by learning (i.e. by engaging in learning events) or by applying certain knowledge and skills (i.e. by performing a task).

Learning Need: A set of learning goals which the learner needs to tackle for performing a certain task. A learning need arises from considering the knowledge and skills required for a certain task, and the knowledge and skills the learner has available. In APOSDLE, the latter is approximated by the learning goals the learner has engaged with in the past.

Learning Path: The learning path means a sequence of learning goals to be acquired in order to optimise the learning transfer for a learner. The computation of the learning path is based on the learning goal prerequisite relation on the set of learning goals.

Learning template: A general structure that refers to a learning goal type and consists of several elements: a) fixed texts that refer to a learning goal, prerequisite knowledge and an engagement activity, b) dynamic slots that are filled with knowledge artefacts, c) input fields, and d) references to co-workers and/or experts.

Prerequisite Learning Goal: We assume that learning goals are (partially) ordered by a prerequisite relation. The prerequisite relation is interpreted as follows: before tackling a certain learning goal, a learner needs to have tackled all learning goals that are regarded as its prerequisites.

Repository Object: a representation of one document retrieved from a back-end system and stored in the APOSDLE System.

Repository: a back-end system accessible by the APOSDLE System by using existing communication protocols and interfaces.

Semantic desktop: The Semantic Desktop paradigm (Sauermann et al., 2005; Decker and Frank, 2004) aims at applying technologies developed for the Semantic Web to desktop computing to finally provide for a closer integration between (semantic) web and (semantic) desktop.

Semantic metadata: Semantic metadata is metadata that stems from knowledge representations. The unique characteristic of semantic metadata opposed to classical metadata is that elements of knowledge representation that are used as metadata can be related by arbitrary semantic relations (Lux et al., 2006).

Task Learning Goal Mapping: A mapping which assigns to each task all skills and elements of knowledge (learning goals) which are required for a successful performance of this task and which are in the learning domain.

Task: In the context of work-integrated learning, a task is an activity which has to be accomplished within a certain period of time, and which requires specific knowledge and skills to be performed.

Task-Learning Goal Structure: The collection of all knowledge states and their associated task representations.
**User Context:** In our understanding context is the substrate in which events occur and which allows a meaningful interpretation of data. Context is characterized by a relevant subset of all surrounding potentially dynamic (e.g. temporal, environmental) information and external and internal conditions. To assess the relevance of information a goal must be considered, which is bound to a knowledge worker. The context is characterized by a potentially dynamic subset of all environmental features. Within APOSDELE, we restrict the information considered for the User Context to those aspects where the acquisition of this information in an IT-based model is feasible. The user context is implemented in the user profile\(^{26}\).

**User Profile Service (User Profile Manager):** The user profile service (UPS) is the software component, which stores and maintains user profiles. The UPS offers user-focused services to the other parts of the APOSDELE Platform. The UPS makes four different types of services available to the platform: logging services, inference services, production services and control services\(^{27}\).

**User Profile:** A user profile is a data structure, which represents an individual users' personal history and current context with respect to the user's work-, learning and collaboration-related experiences. While the user context is a conceptual formal model, the user profile is the data structure that implements parts of the user context. A user profile of a given user consists of four different types of data: user data, environment data, usage data and inferred data\(^{28}\).

\(^{26}\) [https://aposdle.itc.it/glossary/index.php/User_Context](https://aposdle.itc.it/glossary/index.php/User_Context)

\(^{27}\) [https://aposdle.itc.it/glossary/index.php/User_profile_service](https://aposdle.itc.it/glossary/index.php/User_profile_service)

\(^{28}\) [https://aposdle.itc.it/glossary/index.php/User_Profile](https://aposdle.itc.it/glossary/index.php/User_Profile)