Project no. 027087

TENCompetence

Building the European Network for Lifelong Competence Development

Project acronym: Integrated Project TENCompetence

Thematic Priority: 2.4.10

D7.2 Compilation of internal deliverable outcomes ID7.3-ID7.10

Due date of deliverable: 31-05-2008
Actual submission date: 11-07-2008

Start date of project: 01-12-2005  Duration: 4 years

Gottfried Wilhelm Leibniz Universität Hannover

Version 1.0

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
</tr>
<tr>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission</td>
</tr>
</tbody>
</table>

| Services)                                        |                |

<table>
<thead>
<tr>
<th>Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
</tr>
<tr>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission</td>
</tr>
</tbody>
</table>

| Services)                                        |                |
Project Deliverable Report

D7.2 Compilation of internal deliverable outcomes ID7.3-ID7.10

Work package
WP7 – Competence Development Programs

Task
Task 7.1-7.7

Date of delivery

Code name
D7.2             Version: 1.0

Type of deliverable
Report

Security (distribution level)
Public

Contributors
Eelco Herder, Philip Kärger, Daniel Olmedilla, Davinia Hernández-Leo, Sandra Gilabert, Fabien Girardin, Ayman Mognieh, Adriana Berlanga, Hendrik Drachsler, José Janssen, Marco Kalz, Hans Hummel

Authors (Partner)
UHANN, OUNL, FBM-UPF

Contact Person
Eelco Herder (UHANN)

WP/Task responsible
Eelco Herder (UHANN)

EC Project Officer
Mr. M. Májek

Abstract (for dissemination)
Summary of the results of TENCompetence WP7 during the first 30 months. Progress towards a learning path description. Background research and tools for curriculum planning and authoring. Description of navigation and positioning services

Keywords List
Graphical PDP Planning Tool, Hybrid Personalization Service, Positioning, Navigation, Preferences, Curriculum Planning, Learning Path Description

TENCompetence Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 45 5762624 – Fax: +31 45 5762800
Table of Contents

EXECUTIVE SUMMARY ...............................................................................................................................1

1 INTRODUCTION ..................................................................................................................................2
  1.1 THE BROADER PICTURE ..................................................................................................................3
  1.2 OVERVIEW OF THIS DOCUMENT ..................................................................................................5

2 GRAPHICAL PDP PLANNING TOOL ...............................................................................................7
  2.1 INTRODUCTION ...............................................................................................................................7
  2.2 RELATED SYSTEMS ........................................................................................................................7
    2.2.1 LAMS ......................................................................................................................................7
    2.2.2 Moodle .................................................................................................................................8
    2.2.3 RELOAD Learning Design Editor ..........................................................................................9
  2.3 CONCEPTUAL DESIGN ................................................................................................................10
    2.3.1 General use case ..................................................................................................................10
  2.4 SYSTEM OVERVIEW ....................................................................................................................11
    2.4.1 Exploration Interface .......................................................................................................11
    2.4.2 Filtered visualization ........................................................................................................11
  2.5 INTERACTION DESIGN ..............................................................................................................15
    2.5.1 Initial visualization: Explore the proposed learning plans ................................................................15
    2.5.2 Drag-N-Drop Selection of UoLs ..........................................................................................16
    2.5.3 Planning and Exploration ..................................................................................................16
  2.6 IMPLEMENTATION DETAILS ......................................................................................................17
    2.6.1 System Overview ................................................................................................................17
    2.6.2 System Architecture ..........................................................................................................17
    2.6.3 Client side ...........................................................................................................................20
  2.7 FUTURE WORK ............................................................................................................................21
    2.7.1 Concept filtering ..................................................................................................................21
    2.7.2 Widgets ................................................................................................................................23
    2.7.3 The power of defaults ........................................................................................................24
  2.8 REFERENCES ................................................................................................................................24

3 THE HYBRID PERSONALIZER ........................................................................................................25
  3.1 POSITIONING SERVICE .................................................................................................................27
    3.1.1 Implementation ....................................................................................................................28
    3.1.2 Evaluation ..........................................................................................................................28
    3.1.3 Web Service API ................................................................................................................28
  3.2 NAVIGATION SERVICE ...............................................................................................................28
    3.2.1 Implementation ....................................................................................................................29
    3.2.2 Evaluation ..........................................................................................................................29
    3.2.3 Web Service API ................................................................................................................29
  3.3 PREFERENCE-BASED PERSONALIZATION SERVICE ....................................................................30
    3.3.1 Evaluation ..........................................................................................................................30
  3.4 CURRICULUM-BASED PERSONALIZATION SERVICE ...................................................................31
  3.5 HYBRID PERSONALIZER – AN INTEGRATED PERSONALIZATION SERVICE .................................32
    3.5.1 The PersonalizationService interface ................................................................................33
    3.5.2 The AtomicPersonalizationService interface ......................................................................33
    3.5.3 The Wrappers for the atomic services ..................................................................................33
    3.5.4 The Learning Path Provider interface ..................................................................................34
    3.5.5 The configuration framework .............................................................................................34
  3.6 DISCUSSION ..................................................................................................................................35
  3.7 REFERENCES ..................................................................................................................................35
Executive Summary

In this deliverable we present a graphical tool for planning Personal Development Programs, the underlying Hybrid Personalization Service – which combines four complementary personalization mechanisms – and the evaluation of the Learning Path Specification, which specifies the metadata of and structuring elements between learning activities.

Software
The software is available via the TENCompetence SourceForge page at: http://sourceforge.net/cvs/?group_id=159487.

All WP7 software is to be found at: http://tencompetence.cvs.sourceforge.net/tencompetence/wp7/

The Graphical PDP Planning Tool is available at: http://tencompetence.cvs.sourceforge.net/tencompetence/wp7/PlannerTool/.

The Hybrid Personalization Service can be downloaded from: http://tencompetence.cvs.sourceforge.net/tencompetence/wp7/HybridPersonalizer/.

Documentation
Hybrid Personalization Service: http://dspace.ou.nl/handle/1820/1279.
1 Introduction

During DIP1 and the start of DIP2, WP7 has evolved to a work package that focuses on three interrelated tasks. All tasks are concentrated on the creation of models and tools for the authoring, editing, maintaining and using of competence development programs.

Whereas most current standards and tools for e-learning provide relatively sophisticated functionality for the management of units of learning, only little support is provided on the level of curricula. Universities and other educational institutes do have overviews of the courses that are given, or provide standard curricula. However, if a learner has goals that are somehow not standard, she needs to resort to the course descriptions and make a planning on her own. Mentors as well have little support for providing advice and need to resort to their experience. This problem becomes even more apparent if we leave the structured environment of (higher) education and concentrate on corporate learning and lifelong learning.

In corporate and lifelong learning, employers and employees can choose between potentially many course offerings, varying in:

- the competences that they provide
- the level of knowledge and skills that one learns (from beginner to advanced)
- the domain in which they are applicable
- the nature of courses and examination (from formal learning to informal workplace learning)
- availability, schedule, planning, costs
- etcetera.

Within TENCompetence we call these kinds of curricula competence development programs, to indicate that they need not be (predefined) curricula and need not consist of units that were envisaged as courses (spending an afternoon with a colleague, or reading a book on a certain topic might do the trick as well).

The amount of available learning activities and their scopes may be overwhelming – in particular if several institutions combine their offerings (regional, national, international). For course designers this implies that they need to indicate in which situations and for what goals their offerings are suitable. For curriculum designers this implies that they need to interrelate courses, offer several alternatives, indicate benefits and drawbacks, create a realistic time schedule for the potential learners. For learners this implies that they should be advised on the learning possibilities that match their current competence level and that work toward their desired competence level (learning goals), taking into account their restrictions and preferences. In order to accomplish this, we need editors, visualizations and interactive tools to work with a large amount of offerings. Further, selection and structuring mechanisms (we call them positioning and navigation) should be available to find the right stuff in the huge basket. As these tools rely on the underlying (meta)data, we need standardized descriptions of learning activities and their role within competence development programs (CDPs).

In this document we describe the current state of our research, our current visions and our plans to move toward turning these visions into concrete tool.
1.1 The Broader Picture

The aim of our research and development activities is to provide tools for supporting stakeholders in the field of lifelong learning in their activities related to the design, creation, selection, personalization and usage of competence development programs. One issue is that we do not exactly know what these tools should look like, as support for the creation of curricula is lacking in virtually all e-learning systems. Apparently, this is something that still is being done (or rather needs to be done) by hand. There are some good reasons why:

- various failed attempts in the field of adaptive educational hypermedia have shown that you can’t predict with 100% certainty the learners’ goals
- as one almost always needs to trade-off, a system just can’t come up with the curriculum; best it can do, is to provide several options
- the process of creating a CDP, based on an initial idea of a goal, helps in making the goal more concrete. The further one is in the process, the more context one has and the better one can decide whether a learning activity is relevant/fun/interesting
- curricula are not just planned completely beforehand
- “with automated planning, one would get a curriculum that one asked for, not one that one needs”.

At schools or Universities, students are offered curricula that typically consist of a fixed, obligatory part and some space that they can fill in themselves. In practice, students revise their study plan each year, based on their past year’s experiences, available courses and personal factors, such as time constraints, focus in study goals and preferences. Often they also discuss their choices with their peers and with their mentors, who have to approve the plan. In lifelong learning, we see the same effect. On the one hand, many companies offer fixed, possibly obligatory, training programs (we have observed this in the digital cinema case, reported in D7.1). On the other hand, lifelong learners plan their learning activities based on current, possibly short-term needs (‘I need to learn how to work with PhotoShop for this project’), career planning and personal interests (the latter often also in the learner’s private time).

Which learning opportunities learners take up, depends on a large number of factors. Apart from current training needs and wishes, factors as balance between private life and work, current offerings in the neighborhood or region, support from their chefs and personal motivation play an important role. Instead of (automatically) generating a supposedly ‘perfect’ competence development program, we need to support the learners in planning their learning activities themselves.

For this reason, we need to provide the different kinds of users with rich, interactive tools that allow them to construct, inspect and reflect on their competence development programs. The Graphical Personal Development Plan (PDP) Planning Tool adopts a very visual approach that allows learners to create, interact with and evaluate alternative development plans. The learning activities in the user interface are visualized as ‘newspaper advertisements’, which are generated from the learning activities’ metadata. The initial suggested plan (as suggested by the personalization system, explained later on in this section) is displayed as a ‘learning path’ from the middle bottom to the middle top part of the screen. On both sides of the learning path – in the periphery – learning activities that are deemed second choice by the system (but that actually might be deemed first choice by the learner) are displayed. The learner can inspect the details of each activity (who provides the course, what are the costs, is there some form of examination or
certification, is it just online or does it involve class-based activities, etcetera) and add or remove it to her personal development plan.

The whole selection and ordering process should be as tactile as possible, giving the learner the impression that she actually moves, interprets and marks learning activities until the table full of advertisements matches her needs. At any point, the learner may decide to revise her plans by stepping back (or renewed) into the process.

The graphical PDP planning tool heavily depends on a hybrid personalizing service, which consists of four independent personalization services that are combined in a flexible manner. The vertical order of the learning activities is based on the algorithmic curriculum planner and the navigation service. The algorithmic curriculum planner analyses dependencies between learning activities in terms of prerequisites; if a learning activity B requires competences addressed in learning activity A, learning activity B will be placed higher (further from the learner’s initial position) than object A. The ordering is further refined by the navigation service – a collaborative filtering technique that analyses the order in which most users follow the learning activities. The horizontal order of the learning activities is based on their relevance: the more to the middle, the more relevant the objects are deemed to be by the positioning service and the preference-based selection service. The positioning service is a bottom up technique based on latent semantic analysis (LSA). By comparing the content of learning activity or UoLs with the content of learner portfolios, an estimation of the relevance of the learning activity is made. The preference-based service rates to what extent the learning activity matches the learner’s constraints and preferences. As a result, the suggested learning path flows from the bottom part of the screen via the middle of the screen to the top. On both sides of the learning path additional learning activities may appear, which are only sideways relevant and/or do not match the learner’s preferences sufficiently.

Both the personalization services and the graphical PDP planning tool rely on availability of metadata of and structural relations between learning activities. These are provided by the learning path specification, which we presented in D7.1, and which has been evaluated in this project cycle.
Figure 1: Overview of the PDP Planning Tool, Personalization Services and Learning Path Description

1.2 Overview of this document

In chapter two we present the Graphical PDP Planning tool, which is the tool that combines all WP7’s efforts. After an introduction and related work, we describe the conceptual architecture, interaction design and implementation details. This chapter ends with a listing of future work.

IDs reported in chapter 2:
ID 7.4: Specifications Report based on analysis of current practices and user needs for selecting or creating competence development programs
ID 7.5: Evaluated user interface designs for selecting or creating CDPs

In chapter three we present the Hybrid Personalizer and its constituent parts: positioning, navigation, preference-based search and algorithmic curriculum planning. The Hybrid Personalizer combines the bottom-up and top-down techniques in a flexible and innovative manner. We reiterate the functionality of the personalization services and describe their interfaces. The chapter ends with a description of the Hybrid Personalizer, its functionality and how the services are defined.

IDs reported in chapter 3:
ID 7.6: API definition to be delivered to WP3
ID 7.7: Validated service for selecting or creating CDPs
ID 7.8: Validated positioning service, first release
ID 7.9: API definition to be delivered to WP3
ID 7.10: Validated positioning service, second release
ID 7.11: Validated navigation service
In **chapter four** we report on the activities carried out for evaluating the learning path description, which forms the basis for the structuring elements and metadata belonging to the learning activities.

**IDs reported in chapter 4:**

ID 7.3: *Refined version of the learning path description, suitable to be brought as input to standardization bodies*

The concluding **chapter five** summarizes our activities and presents the work to be carried out in the next project cycle of TENCompetence.
2 Graphical PDP Planning Tool

2.1 Introduction

The design and use of interactive information visualization tools, such as e-learning editors, have been widely studied in the past – see section 2.2 for selected related work. As advised by (Shneiderman et al, 2000) visual tools should be designed to be both displays and search tools at the same time.

Some visual schemes generate only one view per information space, but allow the user to zoom in and out, rotate, or in general change his/her own viewpoint on the image resultant from the visualization. This approach to visualizing information spaces inhibits searching and browsing by making it difficult for users to isolate, identify, and analyze parts or aspects of the information space. Users should be allowed to customize and control how the tool at hand addresses information spaces. Moreover, users should be able to specify which part of the information space to visualize in a dynamic manner, making browsing are re-querying information spaces a process of moving between different views and viewpoints at the same time. The latter approach is not only based on the fact that tools should allow free browsing, but also on the general need of users to identify relations within the information space and between information spaces as well.

The efficiency of tools directly derive from the ability of humans to assimilate them and work around them, with these applications and schemes tailored in respect to the human cognitive process and taking account its limitations and powers, designers can hope to maximize their utility. A visualization that overwhelms human sensors will only frustrate its users, who will become largely prompt to erroneous behavior and discontinuity with the information’s context. The failure to take human physiological properties into consideration may be the explanation behind the failure of many complex information schemes to achieve high usability levels.

In this chapter we describe the graphical PDP planning tool, which is the result of several iterations, some of which have been described in D7.1. To put the system a bit more into context, in section 3.2 we describe a number of related systems. The chapter then continues with the conceptual design, interaction design and implementation details. We conclude with recommendations for future work, parts of which will be carried out in the next phase of the project.

2.2 Related Systems

This section summarizes existing tools. We analyzed these tools in order to extract requirements for a tool supporting a learner or a teacher to design learning paths.

2.2.1 LAMS

LAMS\(^1\) (Learning Activity Management System) is a system for creating and managing sequences of Learning Activities. Its authoring tool allows teachers to create and modify sequences of learning activities and store these in the sequence repository), and monitoring

\(^1\) http://www.lamsinternational.com
(where a teacher can select a sequence from the sequence repository, assign a group of learners, activate the sequence for learners, and then monitor their progress). Sequences are kept basic, as the unique relation between Learning activities is a transition relation. Therefore, no sense of requirements or bidirectional relations is carried in the sequence of activities. Moreover, the tool keeps a limited set of types of activities and resources in its toolkit menu. However, the tool enables to set optional activities and offers a preview mode to get a better overview of what is being constructed.

**Figure 2: Basic sequence in LAMS**

### 2.2.2 Moodle

Moodle\(^2\) is a Course management system (CMS) for producing Web-based courses. A teacher has full control over all settings for a course and its management, as well as a flexible array of course activities, such as forums, quizzes, glossaries, resources, choices, surveys, assignments, chats and workshops. The activities are ordered according to the course format by week, by topic or in a discussion-focused social format. Due to its socio-constructivist inspiration, the system nurtures a strong community aspect enabling participants to be aware of each other and collaborate. A path in Moodle consists of a set of phases within a course. However, it is not possible to mix a phase in one course in another course.

\(^2\) [http://www.moodle.org](http://www.moodle.org)
2.2.3 RELOAD Learning Design Editor

As part of the RELOAD\(^3\) project, Phillip Beauvoir and Paul Sharples of the University of Bolton have developed the Learning Design Editor. The editor supports the full IMS Learning Design specifications for Levels A, B and C. In a project manager view, learning planners can organize their Learning Designs. This tool does not carry a very intuitive sense of creating learning paths and sequencing activities, which is probably due to its goal of staying at very close to the IMS-LD complex specifications.

---

\(^3\) [http://www.reload.ac.uk/](http://www.reload.ac.uk/)
2.3 Conceptual Design

2.3.1 General use case

Selecting a learning path is an iterative process that changes over time, depending on the learner's interests. The process starts with the learner entering into the system with initial learning goals. Based on these goals, an initial selection of learning activities is presented to the user – with each learning activity’s location determined by its relevance and position in the tentative learning path. The learner can explore the available learning activities, read the contents and reorder the learning activities to better match her preferences. The learner can include or exclude learning activities from the learning path. This plan can be reviewed and refined in several iterative steps – at any point in time.

2.3.1.1 Goal Definition

The process starts by defining the learning activities around a competence. The reference target is the learning path toward the competence. Just in a few cases, learners start the goal definition from scratch. Usually, they have a general idea and they refine it during the process of preparing their learning path.

Currently, we assume that the general idea of the learner goal is given. As can be read in the concluding chapter 5, we will design and implement a tool for competence matching on job profiles. This tools allows the learner to compare her competence profile (which is part of her ePortfolio) with the requested competence profiles, as listed in job advertisements. The identified competence gap will be used as the initial learner goal.

2.3.1.2 Exploration

As explained in more detail in 2.4 and 2.5 and motivated in the introductory chapter 1, the second step is exploring the available learning activities. Even though the planning tool will provide an initial recommended learning path (by means of the Hybrid Personalization Service, described in chapter 3), it is unlikely that this learning path will fit all (articulated and not articulated) needs by the learner. Even more so, by exploring the available options and reading more details on what these activities comprise, the learner will most likely get a more refined picture of the learning goal. We consider exploration and rethinking as essential elements of CDP planning – in contrast to earlier systems that tried to generate a ‘perfect curriculum’.

2.3.1.3 Review and Refinement

In addition to the exploration interface, the right part of the screen is reserved for the actual learning path – the selection of learning activities that the learner would like to include in the plan. At each point in time, the learner can add more activities, remove one or more activities or reschedule activities within the learning path. The metadata of the learning activities (the more detailed information, as shown in figure 5) helps the learner in making sound decisions.

2.3.1.4 Goal Completion and Refinement

Once a learner is happy with the learning path, or wants to continue the planning at a later point, the learner can save the current planning and continue at a later time. The goal might be reached for now, but the actual goal is a running target – which might change each time the learner gains new experiences (a student typically refines her study plans at the start of a new semester, as she needs to actually plan the next activities).
2.4 System Overview

2.4.1 Exploration Interface

The user requirements suggest an interface for information exploration that is simple and easy to use. With the aim of complying with these guidelines, the interface uses the metaphor of bubbles to encapsulate a range of subjects. Each bubble (object) represents a unit of learning and contains all information related to the Units of Learning (UoLs or ‘learning activities’) it represents: title, description, prerequisites, etc. Each UoL bubble includes visual information that allows an easy identification. **UoL topic area:** Each bubble has a different color assigned (red, green, blue, etc) depending on the UoL topic area. **UoL size:** Bubble size indicates the number of credits per UoL.

The approach of using bubbles as a metaphor to represent information emerged in 2002. Groxis and Cloudmark argued that information can be modeled as a kind of node, or bubble, and that related bubbles can be nested one inside another. These interfaces also rely on color cues based on the category, importance, or urgency of information to make navigation easier. This metaphor has been selected because of its flexibility, ease of use and ability to represent abstract information, as well as various types of data (text, images, video). The same kind of metaphor is applicable in different contexts while planning learning goals.

2.4.1.1 Details-on-Demand: UoL Tooltips

Each bubble includes a subject description displayed as a “tooltip”. The description summarizes the UoL contents, allowing the users (learners) to quickly grasp the subject gist. The description includes a generic image that is automatically generated based on the UoL short title. These images, together with spatial and visual memory allow for a faster recall.

![Figure 5: Interface: each UoL includes a description, credits, difficulty, topic area, and a picture that helps to describe it](image)

2.4.2 Filtered visualization

The interface collects information and updates the visual space according to clusters (UoL bubbles) based on the properties identified by the system (provided by the database and API, WP5 services). The interface also displays conceptually related subject topics to the user goals.
The figure below shows how the term bubble is used to filter the information presented to the user in order to avoid overloading the interface with visual elements.

![Filtered visualization: reduces information overload](image)

**Figure 6: Filtered visualization: reduces information overload**

### 2.4.2.1 Interface Layout

The interface is divided into two areas: an exploration area with the proposed learning path, and a personalized area where learners can configure their own learning path. Learning paths are organized by a “temporal line” as to provide cues on the path duration to the learners. This temporal organization ultimately depends on the learner’s goals.

All units of learning associated with the learning goal float around the interface, following a proposed learning path (vertical axis) and organized by relevance to the learning goal (horizontal axis) - as illustrated in Figure 7-10. Units of learning are also linked based on the dependencies among each other.
Figure 7: Interface: suggested learning path (proposed path shown)
Figure 8: Interface: Units of learning are linked based on dependencies among each other.

Figure 9: Interface: Learners can drag each UoL to the desired position on the user's planning area.
Figure 10: Interface planning area: Learners can create or redefine a customized learning plan based on their own criteria by organizing each UoL according to their preferences.

2.5 Interaction Design

This section shows a typical use case of a bubble-based interface for UoLs. The use case is based on the analysis of user requirements. That is also the reason why this section could also help users in their familiarization with the visualization and interactions of the interface.

2.5.1 Initial visualization: Explore the proposed learning plans

The user is presented with two areas: suggested learning paths according to the positioning service and planning area for composing the personal learning path.

The suggested learning path visualizes related UoL organized as bubbles with dependency relations. These dependency relations are shown only when the UoLs are selected. The UoLs are also somehow clustered according to the proximity suggested by their content. Their position in terms of the X and Y-Axis is determined by the hybrid personalization service (see chapter 3). The size of the bubble is the number of credits (“cost” in terms of time) that is estimated to satisfactorily complete the UoL. The difficulty level is illustrated by means of intensity.

Moreover, when a bubble is selected, a media visualization associated to the UoL is shown. The media visualization can be text combined with an image (and eventually also a video).
there are several media visualizations associated, they are shown successively. When the bubble is no longer selected, the media visualization disappears. The bubbles can be redistributed by the user while exploring the area with the suggestions.

2.5.2 Drag-N-Drop Selection of UoLs

Drag-n-Drop allows the movement of objects (UoLs) in the interface. In this case the selected UoLs (from those available in the exploration area on the left side part of the screen) can be moved to the area where the actual learning path is planned the right side part of the screen. The movement process follows the standard Drag-N-Drop paradigm: upon hovering the UoL, the user presses and holds the main mouse button, moves the mouse to the destination and releases the UoL at its new location. The Learning Path can contain multiple learning activities, which are vertically organized according to the time at which the will be followed. Multiple UoLs can be added to the personal plan organized according to a temporal line.

2.5.3 Planning and Exploration

After several iterations (dragging and dropping bubbles, exploring the different UoLs opportunities suggested to achieve the learning goal) users can easily personalize their own learning paths.
2.6 Implementation Details

Initially, the PDP tool was planned to be implemented in Java. A very primitive prototype was developed using this programming language, but during the last few months the implementation requirements changed, and it was agreed that the development should be conducted in Flex. Hence, the expressiveness of the tool could be greatly facilitated. As a first approximation, we have employed Flash to build a prototype that meets the new requirements.

As mentioned in other sections, the planner aims at providing users with a highly interactive and expressive representation of the information generated by the positioning and navigation services, which are implemented in Java. Thus, we had to design a system that enables the communication between both programming languages (Java and Flash). We have addressed this problem by using Java servlets as an interface between the personalization service engine and the Flash web client.

2.6.1 System Overview

Users enter their personal preferences and their learning goal through the planner. According to these parameters, the client requests a recommended learning plan from the personalization services, which retrieve from a MySQL database the information associated with the input data, and compute the most optimal learning path. For each learning activity contained in the plan, its location on the screen is also calculated by the engine. Finally, the planner visualizes the resulting data, which is vertically arranged by time, and horizontally centred by their distance to the user’s learning goal and personal preferences (for more details, see chapter 3). At this level, learners are expected to explore, search, compare and select those learning activities that are relevant to their academic goals.

The SQL database currently contains just one competence development program (the Introductory Psychology Program of the Open University of the Netherlands), plus a default user profile and learning goal. Therefore, the PDP tool just concentrates on requesting the existing data, and provides the user with a highly expressive and interactive visualization of the available information.

2.6.2 System Architecture

As shown in figure 11 (architecture), the Flash Web Client (flash player v8) establishes a connection with a Tomcat Apache server (apache-tomcat-6.0.14), where the servlet is hosted. The learning path plus the description of the learning activities contained in it are requested through a GET message. After this, the servlet invokes a Java framework that communicates the petition to the personalization engine. Once the computation of the learning path is finalized, the personalization engine sends the resulting information to the Java framework, who assigns learning categories and related images (keywords) to the learning activities. Finally, the servlet collects a Java object that encapsulates the result of the whole process, and translates it into an XML file, which is sent back to the Web Client. Figures 12 and 13 show an example of the XML files describing the learning path, and learning activities metadata.

The image attribute of the metadata XML file is used by the Web client as a keyword to retrieve an image from the database, which is a visual description of the learning activity. On the other hand, the category field groups learning activities that are semantically related. The web client displays the categories in different colours.
Figure 11: System Architecture
<?xml version="1.0" encoding="iso-8859-1" ?>

- <UoLsInfo>
  - <UnitOfLearning>
    <ID>22</ID>
    <FullName>De biologie van het gedrag</FullName>
    <ShortName>Biologie</ShortName>
    <Description>De biologische psychologie is de wetenschappelijke bestudering van de biologische basis van het gedrag en mentale processen van mensen. Echter, ook de resultaten van dieronderzoek (dat strikt genomen onderdeel vormt van de neurowetenschap) vormt een bela</Description>
    <lang>nl</lang>
    <teacher>Miller</teacher>
    <cost>5.0</cost>
    <difficulty>2</difficulty>
    <postcondition>Biopsychology Overview</postcondition>
    <prerequisite />
    <locationX>-5.0</locationX>
    <locationY>0.0</locationY>
    <Category>Biology</Category>
    <Image>molecular</Image>
  </UnitOfLearning>
  - <UnitOfLearning>
    <ID>35</ID>
    <FullName>Persoonlijkheid</FullName>
    <ShortName>Persoonlijkheid</ShortName>
    <Description>De persoonlijkheid is het geheel aan eigenschappen die men aan een levend wezen kan verbinden. Het geheel van deze eigenschappen vormt het wezen zijn persoonlijkheid, die men kan karakteriseren in bepaalde termen die hanteerbaar zijn op dat vlak. Zo kan</Description>
    <lang>nl</lang>
    <teacher>Miller</teacher>
    <cost>5.0</cost>
    <difficulty>2</difficulty>
    <postcondition>The Individual</postcondition>
    <prerequisite>Social Psychology Foundations</prerequisite>
    <locationX>-5.0</locationX>
    <locationY>20.0</locationY>
    <Category>Human behavior</Category>
    <Image>child</Image>
  </UnitOfLearning>

Figure 12: XML description of the learning activities metadata
2.6.3 Client side

The Flash client is responsible for showing the information received from the server in a convenient way, and provides the interactivity needed for the use of the application by the user.

The interface communicates to the server the request for the XML document of a given learning path. The returned XML, as mentioned before (fig. 2 and 3), includes the general structure of the studies as well as the detailed information of every unit of learning.

In the interface, a unit of learning is mapped to an ActionScript 2.0 Node class that is linked to a graphical object, which is rendered as a circle containing the short name of the learning activity. This object contains all the information related to the learning activity, and enables the interaction between the user and the unit of learning (it answers to events like click, hover, or drag). The graphical characteristics of the circle are parameterised by some of the attributes of the learning activities. For instance, the dimensions of the circle are given by the cost of the activity, whereas colour depends on its category.
The interface creates internally two node groups: one for the proposed plan, and another for the user one. All the learning activities retrieved from the XML are added to both groups. However, all the nodes in the learner plan area are initially inactive (not visible). When the user drags a unit from the proposed plan (left area) to the learner plan (right area), the dragged node is activated in this area, and hidden from the other.

Each node group keeps a link list between the different units of learning. The dependency relations among objects are mapped to an ActionScript 2.0 graphical object (an arrow).

All this data is managed by another ActionScript class, the Manager class, which establishes the connection with the Tomcat server, requests the data to be displayed, and controls user’s interaction inputs (dragging, hovering, …).

The prototype of the application is at the following URL:
http://upf.ernestoarroyo.com/Planner/, and the dynamic XML is at http://planner.ernestoarroyo.com:8080/Planner/MetadataInfo.

2.7 Future Work

As described in more detail in the last chapter of this deliverable, we plan a qualitative user study, to be carried out at the UPF at the start of the next study year. The main aim of the study is to solicit feedback for future iterations and to support our claims on the benefits of this approach.

Apart from the results of the study, we see several directions to follow during the further development of the graphical PDP planning tool. In its current state, the Graphical PDP Planning Tool offers quite some functionality that supports exploration and selection of learning activities. However, it is desirable to provide the learner with more tools to get different perspectives on the available learning activities and her current and past interaction history – for recapitulation of choices, for remembering why certain activities were included in the learning plan, etcetera. For this, we consider several filtering mechanisms and widgets to be included in the interface. Further, we plan to use empirical data for creating suitable default values, which is essential for a smooth user interaction. These directions are explained in more detail in this section.

2.7.1 Concept filtering

In order to reduce cognitive overload, we propose a universal system of filters integrating the functions required by the learners. These filters will provide a centralized point of access to UoL information, allowing users to view and explore this information in different ways. A universal filtering scheme will allow the configuration of all elements in the visualization. These filters will allow exploration of UoL and their relationships using different levels of granularity (detail).

The UoLs (bubbles) that are displayed in the user interface, will be those that the system determines as the most important, based on the learner’s learning goal and preferences. The criteria that determine the UoL spatial arrangement can be configured to show different learning paths and relevant or related UoL.

A visualization filter will allow modifying the display of relations and dependencies between concepts and the level of detail shown. At one level it will show the dependencies between learning activities as a link that visualizes the type of relationship between them. As the filter level of detail increases, new UoL relations can be observed according to the desired/selected
level of granularity. Another type of filtering involves dynamically filtering UoL search results (query).

The resulting selection of units of learning can be filtered using widgets to set the following parameters – which correspond to the metadata in the learning path specification (see chapter 5):

- **UoL type**
  - prerequisite
  - optional
  - topic area
  - etc
- **UoL source**
  - proposed path
  - tagged
  - suggested UoL
- **Difficulty**
- **Number of Credits**
- **Language**
- **Teacher**
- **Intersections with other learning goals**
- **Similarity**
- **Subject area**
- **Other.**

The process of use of filtering UoL information will take place in two steps: pre-configuration, dynamic filtering and the combination of both steps.

All filters will have default values to "optimal", but it will be possible for the learners to *pre-configure* filters according to the type of results/concepts they expect to get before making the search for concepts. The results of the query will meet criteria for filter settings.

*Dynamic* filtering mechanisms will be a central part of the user interaction. As described above, filters can be applied for the initial selection, but they can also be used for providing several perspectives on the units of learning, to make an informed choice (as an example, the learner might first use filters that emphasize the learning goals ‘what would be the smartest choice’, and then to set filters according to her preferences ‘what would be the nicest choice’ to get a feeling for the gap between her wishes and her needs):

- Learner starts by selecting several learning goals that might be of some interest.
- The learner selects interesting or relevant subjects and drags them to the personalized learning path area of the interface.
- The learner explores as many UoL (according to initial results and interests).
- At one point the screen is saturated with information (related UoL), so the filters will be used to de-clutter the screen.
- The learner applies filters to explore UoL according to individual preferences. Filtering by language, number of credits, etc.
2.7.2 Widgets

The user can access the various filter mechanisms via widgets. Widgets are small independent components that communicate with the interface to modify their behavior or visualization. The widgets allow for highlighting or hiding the UoLs that meet a certain criterion set by the learners.

Widgets are present at all times in the periphery of the interface and when activated, present the user different filter settings options. The widgets are also able to interact with one another. Updating information in one of them will be reflected in other widgets. The widgets will communicate with the user interface via a standardized API. In addition, we consider a number of other widgets, which are described below.

2.7.2.1 Session States

One of the main disadvantages of tree-based navigation is that it is not appropriate when large amounts of information are used. Short term memory limitations indicate that users have difficulty using trees as a means of information retrieval when it is necessary to navigate several levels deep or there is a large number of nodes per branch. It is common for users to forget the access point to a branch of the tree, or why they chose a certain branch of the tree. This effect is increased when navigating within recursively tree branches, especially when the nodes have multiple non-hierarchical relationships, as in the case of relations between optional UoL.

This motivates an analogy of states to be used by the system. A state provides pointers that enable users to retrieve the context of navigation as quickly as possible. The system will provide mechanisms to save and restore the state of the search during any point while exploring and planning their learning paths. These states are denominated sessions or bookmarks. This feature will allow either to navigate through the sessions as a history, or between critical points of exploration.

2.7.2.2 History, Bookmarks and Favorites

The interface will provide a history feature that implicitly stores and records the users' activities. The interface will be able to save all steps that are followed during the planning session. Each selection, each search, each filter applied, and so on, will be recorded in the order they are made and may be undone step by step (or all at once), similar to the Undo functionality, as is common in many desktop applications.

The history widget will visualize the interaction trace. By interacting with the widgets, users can undo or redo the steps taken and they can take ‘snapshots’ of the interface at some point in time. Clicking on the screenshot visualization results in restoring the interface to the saved state. States can be stored in the long term by dragging it to the bookmark widget.

The functionality of bookmarks allows users to store the state of the screen at any point in the navigation. Bookmarks are states in navigation where the user feels the need to take a mental photograph or mark the navigation state. Each bookmark is associated with an image that represents the learning path at any given point. The bookmarks will be represented by two different ways:

- A widget of bookmarks next to the interface described by keywords chosen at the time it was generated
- Markers within the bookmark widget that allows browsing bookmarks with respect to time (forward and backwards) in the order they were created.
An important feature is the persistence of the application states between sessions. The widget allows sessions to save the current state of the application, including the steps taken to reach this point, enabling learners to freeze a state and return to it at another time. To load a past session, learners will simply drag the bubble that represents it over the interface work area.

Each learning activity will have icons that will allow users to add the activity to a list of favorite items. This list contains learning activities that – for one reason or another – are relevant or interesting to the learner, but that are not considered essential to the search conducted at that time. The favorites widget will keep a list of UoL aggregates, which may sorted according to the order of entry, by alphabet or relevance. The widget will allow access to these UoL at any time during the interaction with the interface. The bubbles of the favorite UoL may be dragged to the planning area in the interface to redefine the learning path. Dragging bubbles out of the widget will remove them from the UoL favorites. Additionally, a filter will allow users to highlight the UoL that are on the list of favorites, thus making it easier to identify and interact with them.

### 2.7.3 The power of defaults

Even though customization of views and filters is one of the core functions of an information visualization systems, the initial views play an important role in guiding the user and helping the user to interpret what information is visualized, its structures and they way the user can interact with it by manipulating the views and filters. Users depend on defaults to learn how to use the interface and are rarely changed by them. It is important to determine which the interface filters optimal default settings are.

The default filters will be determined in initial evaluations and based on design criteria in such a way to allow learners to have a satisfactory experience and obtain useful results during the first uses of the system. Once the learners have found the limits of the system (with default values), they may explore options (filters) of the interface to obtain the desired results or display information according to their preferences/individual needs.

### 2.8 References


Stuart Card et al. Readings in Information Visualization, Using vision to think. Morgan Kaufmann.

3 The Hybrid Personalizer

In D7.1 we presented four complementary personalization services:

- the **positioning service**, which estimates the relevance of a learning activity by applying latent semantic analysis (statistical comparison of textual contents) on the learning activities and the learning goal. This technique does not require any metadata to exist, but has the disadvantage that it inherently introduces some uncertainty.

- the **navigation service**, which uses collaborative filtering techniques for determining the most popular followed steps after having completed a learning activity; similar to the positioning service, the navigation service does not require any metadata and makes use of the ‘wisdom of the crowds’. As a disadvantage, this service requires a relatively large user base and might not properly take envisaged didactics into account.

- the **algorithmic curriculum planning** imposes an order on learning activities, by comparing their prerequisites and learning outcomes; if a learning activity B requires competences that can be learned from learning activity A, B is placed after A. This top-down technique requires metadata in terms of prerequisites on the learning activities (which is an authoring effort), but is able to generate suggested paths that follow didactic principles.

- the **preference-based selection of learning resources** (also known as ‘Skylining’) analyses to what extent a learning activity matches the learner’s preferences (which can be anything ranging from type of assessment and study load to where the course is given and the costs associated with it).

![Hybrid Personalizer Architecture](image)

Figure 14: Hybrid Personalizer Architecture
As explained in the previous chapter, these complementary personalization services have been combined into a hybrid personalizer, which allows a flexible combination of the personalization outcomes. The service relies on wrappers that provide a standardized interface to the independent services and that provides the metadata, derived from the learning activities (‘Units of Learning’).

The graphical interface does not directly access the outcomes of the services, but rather simply requests the Y-Axis location (where in the learning path the learning activity should be placed) and the X-Axis location (how relevant is the learning activity for the learner and to what extent matches it the learner’s preferences).

**Figure 15: Placement of learning activities on the X-Axis and Y-Axis**

In this chapter we provide further technical details on the four personalization services (which have been introduced in D7.1) and how they have been combined in a hybrid personalization service.
3.1 Positioning Service

The Positioning Service builds on the assumption that prior learning can be approximated through an analysis of the similarity between learner portfolios and (target) learning material. For this purpose Latent Semantic Analysis is used to calculate the similarity. The concept of the service is shown in figure 16 (Kalz et al 2008).

![Figure 16: Concept of the Positioning Service](image)

The data layer is responsible for preprocessing the available learning content, ePortfolio content and general language corpus documents. In the Similarity Measurement Layer a latent semantic space is build in which the query content, in this case the ePortfolio documents, is projected in. The result of the service is a list of learning content units/ learning activities with cosine measures. The output layer produces this list with learning activities and cosine values, while specific thresholds can effect the output list.

The Positioning Service is located in the bottom up cluster of the Personalization Pipeline. In this pipeline the positioning service analyzes portfolio data and projects them into a latent semantic space build from general language data, domain data and the content related to learning activities in learning networks. The result of the positioning service is used by the personalization pipeline to recommend a set of learning activities which fit to the learner’s preferences and prior knowledge. In this case learning activities with a very high cosine similarity are omitted for this learner.
3.1.1 Implementation
The Web service builds on the Php Lsa Engine, which is packed with an Easy-Php-Server including everything needed to run the webservice (Apache, MySQL, LSA Engine). The LSA uses the ALGLIB library (Bochkanov & Bystritsky, 2008) for the Singular Value Decomposition. The performance of the Positioning Web Service has been evaluated and optimized by external domain experts on a data set from psychology. We expect that the service needs to be optimized with training data for every implementation. From a technological perspective the web service has been tested regarding reliability and performance. The separation of the construction of the latent semantic space and the query process has improved performance for the service a lot. But the query process takes still around 30 seconds for a test dataset of 800 documents until the service is able to present results (total size 7 MB). This depends to a large extent on the corpus and query size. A demonstration for the Web Service is located at http://145.20.132.193/consum.php

3.1.2 Evaluation
We conducted a case study in an introductory psychology course at the Open University of the Netherlands. Students were asked in advance to build a dossier-type portfolio of products they produced in their past education or work context. We used Latent Semantic Analysis to analyze the similarity between the students’ documents and the content in the learning activities of the course. Domain experts were asked to rate the similarity of documents and to decide about exemptions based on this similarity. A first inspection of the results shows us that the similarity measurement that are produced by the system can differentiate between learners who sent in different material and between the learning activities and chapters.

3.1.3 Web Service API

<table>
<thead>
<tr>
<th>Positioning Web Service Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated similarity between portfolio documents and learning network content</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Method</th>
<th>Description</th>
<th>Input (Parameter)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Position Values</td>
<td>Get</td>
<td>Return a list of UoL annotated with cosine values</td>
<td>Iduser=xx</td>
<td>2-dimensional array of floats. Each UoL with its calculated cosine values.</td>
</tr>
<tr>
<td>Frequency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When visualization tool has to be re-ordered/reorganized.</td>
<td></td>
<td>Iduser = Integer Learning goal = Array (Strings)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 Navigation Service
The Navigation Service is a bottom up service of the Integrated Personalization Service. Therefore, it is part of the Hybrid Personalizer’s bottom up technology cluster (together with the Positioning Service). These are bottom up technologies because they require no top down maintenance of metadata thus they rely on information from the learners. Beyond the specification of Deliverable 7.1, we added an additional function to the service API called getNextStepPropability. For every Unit of Learning (UoL) the Navigation Service
calculates its relationship to other UoLs in the Learning Network in order to provide measures for the Visualisation Tool to place the UoLs on the screen. Therefore, it observes the behavior of the learners and their usage of the UoLs in the learning network. Every successful completion of a UoL is saved in a transition matrix. Based on this matrix the Navigation Service provides a measure between UoLs that represents their relationships between each other according to the learning behavior of the learners.

3.2.1 Implementation
The web service is created with PHP5 on an Apache 2 web server. The transition matrix is saved in a MySQL 4 database. PHP5 has an integrated web service class that enables us to make the Navigation Service’s results available for the Personalization Interface. A demo for the Web Service is available at http://lnx-otecexp-004v.ou.nl/navigation/test/consum2.php
The performance of the Navigation service has been evaluated within an experiment with students of a psychology course (Drachsler et al. 2008). From a technological perspective the web service has been tested regarding reliability and performance with the WP partners.

3.2.2 Evaluation
We carried out an experimental study within the regular “Introduction Psychology” course as offered by the Psychology faculty of the Open University of the Netherlands (OUNL). The participant group consisted of 190 students, split between an experimental and a control group. The experimental group received recommendations. Results show that the experimental group consistently needed less time to complete equal amounts of LAs, which effect was found to reach significance after 4 months. The experimental group from the beginning onward created more personalised learning paths. The qualitative data about satisfaction from the recall questionnaire underlined the quantitative results about the actual use of the PRS.

3.2.3 Web Service API
The Navigation Service API still relies on the specification we made in the deliverable 7.1. For the current prototypical Visualisation Tool we did not implement all functions into the service. For a future integration of the service into the PCM minor adjustments have to be done in order to feed the service with required data from the 10C infrastructure.

In the following we give a specification of the additional function called getNextStepPropability we added to the Navigation Service.

<table>
<thead>
<tr>
<th>Navigation Web Service Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate relationship between UoLs based on behaviour of the learners</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Method</th>
<th>Description</th>
<th>Input (Parameter)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Next Step Propability()</td>
<td>GET</td>
<td>Request a relationship description for a current UoL an additional array with UoLs from the Navigation Service</td>
<td>UoL_ID = xx UoLs=xx</td>
<td>2 dimensional Array of Integers For each UoLs it relationship to the requested UoL_ID is calculated in a range between 1 and 15</td>
</tr>
<tr>
<td>Frequency</td>
<td>DATA Fields</td>
<td>Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whenever the Visualisation Tool have to be reorganised according to decisions of the learner</td>
<td>UoL</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UoLs</td>
<td>Array of Integer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.3 Preference-based Personalization Service

The preference-based personalization service refers to the preferences specified by the learner (cf. Deliverable 7.1) in order to provide meaningful personalized locations of learning activities: preferred learning activities get location numbers close to zero while non-preferred ones get higher values. The service can be configured according to two preference semantics: the qualitative Pareto semantics and a quantitative semantics.

**Qualitative semantics.** Learning activities that are among the first skyline (i.e., Pareto optimal) are given the location zero. The second skyline, i.e., the learning activities that or optimal among all learning activities except the first skyline, are given higher values. This approach corresponds to the qualitative semantics proposed by Pareto and exploited for so-called skyline querying. However, this semantics is very strict. For preferences over many dimensions it is known to include almost all objects, that is, all learning activities are optimal according to high-dimensional preferences. To overcome this problem, we introduced a second semantics in our implementation: a quantitative one.

**Quantitative semantics.** The quantitative semantics computes the location of a learning activity by counting the amount of alternative attribute values that the learner considers better (in terms of the defined preference order) than the actual value of the learning activity. If all attributes of a learning activity are optimal, it is given the location value zero. For each attribute that bears a non-optimal value, this number is increased dependent on the rank where this value appears in the preference order given by the user.

![Figure 17: How preference information is used to distribute the surrogates over the x-axis.](image)

The specification stating which semantics used by the preference-based personalization service is configurable by means of the configuration framework integrated into the Hybrid Personalizer.

3.3.1 Evaluation

We have performed a number of experiments with the lecture database of the learning management system of the University of Hannover. This system currently comprises 9829 lectures. We provided preference-enabled queries and showed how preference-enabled queries
optimize the result set and provides the desired learning resources, without pruning relevant results or returning non-relevant objects.

For more details, see (Abel et al, 2007) and (Kärger et al, 2008).

### 3.4 Curriculum-based Personalization Service

Based on the pre- and post-conditions given for each learning activity, the curriculum-based personalization service is able to induce information about which learning activity is to be visited by the learner at what state of the learning process. For this the service exploits information about the learner’s current knowledge and information about the learner’s learning goal, i.e., the knowledge the learner is aiming at with the current learning process. The information provided by this service for the PDP Planning tool is twofold: first, the curriculum-based service is able to compute a Learning Path (LPath) guiding the learner from her current state to her learning goal; and second, the service is able to compute locations of the learning activity’s surrogates given the amount of learning activities in the learning space forming potential prerequisites for a learning activity.

**Learning Path generation.** Given a set of learning activities, the learner is enabled to request for an appropriate LPath. The computation of the path follows the algorithmic curriculum planning as it described in Deliverable 7.1. For the communication with the graphical user interface, a Learning Path class has been implemented that is used for both, for the intelligent generation of the Learning Path in the service as well as for the visualization in the PDP Planning Tool.

**Curriculum-based computation of locations.** Although the learner may not yet requested for a Learning Path, the service can be used to locate learning activity surrogates in a personalized way. The learning activities that are close to the learner’s current knowledge according to any potential Learning Path are given the location value zero. Based on these most appropriate next steps and by following all potential paths the service is able to generate, the locations for potential follow-up learning activities are computed.

![Figure 18: Computation of locations and generation of a Learning Path out of the preconditions (below the icons) and the post-conditions (above) of the Units of Learning.](image)

For more details, see (Baldoni et al, 2006).
3.5 Hybrid Personalizer – An Integrated Personalization Service

The atomic Personalization Services as they are introduced up to here build up the basis for the advanced integrated Personalization Service called Hybrid Personalizer. Each of the atomic services provide the PDP planning tool with complementary information about which Unit of Learning suits a learner’s needs in her current situation. Combining these complementary input values is the challenge the Hybrid Personalizer is dealing with. The four atomic services can be divided into two bottom up approaches, namely the navigation and positioning Web Services, and into two top-down approaches, namely Preference-based and Curriculum-based personalization.

![Figure 19: Architecture of the Hybrid Personalizer integrating the four Atomic Services (orange boxes).](image)

The atomic services together with the integrated service on the one side and the graphical user interface of the PDP Planning Tool on the other are separated conceptually. That is, java interfaces as well as Web Service interfaces ease the communication between the components and allow for an ad-hoc adoption to new data or systems (e.g., the forthcoming integration into the PCM). Moreover, a configuration component allows a fine-grained tuning and adoption of how the returned values of the atomic services are used to compute a single personalization value. By this means, the strategy of the hybrid personalization can be modified easily.
In the following we provide more detailed but still conceptual information about the Hybrid Personalizer’s implementation. For more details we refer to the TENCompetence cvs WP7 Module and to the Java-doc for the implementation to be found at http://www.l3s.de/~kaerger/HybridPersonalizer/doc/.

### 3.5.1 The PersonalizationService interface

The task of the Hybrid Personalizer is to combine the complementary personalization services in a way that information is generated that is exploitable to place the learning activity’s surrogates in a personalized way. Therefore, we offer the java interface PersonalizationService providing the following methods:

- \( \text{List<Double>} \text{ getLocation(String UoLID, String learnerID)} \) returns a pair of double values. Each coordinate is a values between 0 and 1. This pair represents the location of the learning activity’s surrogate in the Graphical User Interface of the PDP Planning Tool.
- \( \text{List<List<Double>}> \text{ getLocations(List<String> unitOfLearningIDs, String learnerID)} \) returns a list of pairs. The order of the elements in the returned list corresponds to the order of the provided list of Unit Of Learning IDs.

By means of these two methods any component is able to retrieve locations of surrogates that correspond to their appropriateness for the learner is her current situation. In order to call the atomic services in a unique way, we introduced an interface that is implemented by the atomic services or their wrappers, namely the AtomicPersonalizationService interface.

### 3.5.2 The AtomicPersonalizationService interface

Each of the four atomic services deliver their data via arbitrary technology. In order to easy their integration and to allow to simply plug-in potential new atomic services, we provide the AtomicPersonalization interface. It is implemented by the wrapper classes for each of the four services and provides to following methods.

- \( \text{Double getLocation(String UoLID, String learnerID)} \) returns a single double value between 0 and 1. This value represents how appropriate is the given Unit of Learning for the given learner.
- \( \text{List<Double>} \text{ getLocations(List<String> unitOfLearningIDs, String learnerID)} \) returns a list of double values. The order of the double values in the returned list corresponds to the order of the provided list of Unit of Learning IDs.
- \( \text{void init(Set<String> unitOfLearningIDs)} \) initializes the atomic service. This method is also used to update the service if some metadata of the learner or of a Unit of Learning changed. For some services that implement a cache, calling this method may be needed if a new Unit of Learning is introduced.

### 3.5.3 The Wrappers for the atomic services

Each Personalization Service is wrapped by a class implementing the AtomicPersonalizationService interface.

**The class NavigationBasedService** forms the wrapper for the Navigation Web Services. It is implemented as an Axis client and communicates with the actual Navigation Web Service via SOAP. This class scales the values returned by the Navigation Web Service to a value between 0 and 1.
The class PositioningBasedService is the wrapper for the Position Web Service. Similar to the NavigationBasedService class it consists of an axis Web Service client able to communicate via SOAP. Again, this wrapper scales the output values of the Positioning Web Service to values between 0 and 1.

The class PreferenceBasedService is responsible for computing locations based on preference information. It implements the AtomicPersonalization-Interface. The getLocation-method returns a Double value ranging from 0 to 1. This number represents how appropriate the given unit of learning is according to the preferences of the given learner.

The class CurriculumBasedService wraps the implementation of the algorithmic curriculum planner. It exploits the information delivered by the planner to compute numeric values for learning activities. These values correspond to the position of the given Unit of Learning in any learning path that starts at the learner’s current knowledge.

The class FairXValueDistributor. We provided a fifth implementation of the AtomicPersonalizationService, namely the FairXValueDistributor class. This implementation of an atomic personalization service does not take any information about learners and Units of Learning into account. For the computation of a double value location on the x-axis only the amount of Units of Learning with the same y-axis values are considered. The x-values are computed in a way that all these Units of Learning are distributed over the x-axis equally. This is of particular usefulness if no information about learner preferences or about other learner’s steps (as needed for the navigation service) is exploitable for computing an x-axis value.

3.5.4 The Learning Path Provider interface

The LearningPathProvider interface offers two methods that deliver Learning Paths. One method provides a learning path given information about the learner (her current knowledge and learning goal). The other receives a learning path ID and returns the corresponding learning path.

- LearningPath computeLearningPath(Set<String> UoLs, Set<String> knowledge, Set<String> goal) computes a learning path for achieving the competences in the given goal (a set of competences) by assuming a learner’s knowledge as it is defined in the set knowledge (another set of competences).
- LearningPath getDefinedLearningPath(String learningPathID) returns a predefined learning path, for example a previously stored learning path from the TENCompetence server.

Currently, we provide one implementation of this interface called CurrBasedLearningPathProvider. The Curriculum-based learning path provider makes us of the algorithmic curriculum planning tool to compute a learning path.

3.5.5 The configuration framework

Merging the output of the four atomic services in order to provide a single personalized view on the learning space is a challenging task. There are arbitrary many ways of combining the output of the atomic services. This may depend on the metadata available for computing the personalized locations. Dependent on the data available one may also want to put different weights to the results of the atomic services, leave some of them out, or even let the user decide how to configure the Hybrid Personalizer.
3.6 Discussion

Merging the output of the four atomic services in order to provide a single personalized view on the learning space is a challenging task. There are arbitrary many ways of combining the output of the atomic services. This may depend on the information available for computing the personalized locations; if there are no portfolios available for the current learner, the Positioning Service cannot be applied. Depending on the data available one may also want to put different weights onto the results of the atomic services, leave some of them out, or even let the user decide how to configure the Hybrid Personalizer.

Currently, the combination of the four personalization services and their weights is based on 'heuristics and intuition'. We plan to evaluate the graphical curriculum planner with students, who will use the planner for planning their study activities for the upcoming year(s). The observed usage data, as well as transcriptions of the user comments, will be used for an informed configuration of the Hybrid Personalizer.

3.7 References


4 Learning Path Description

4.1 Introduction

The learning path specification that has been developed and reported in D7.1 and is displayed for reference in Appendix 1, has been presented to a wider audience through a publication (Janssen, Berlanga, Vogten, & Koper, 2008) and presentation at the ePortfolio 2007 conference (Janssen, Berlanga, Vogten, & Koper, 2007). Besides, it has been brought to the attention of the Dublin Core Education Application Profile Task Group. They were very receptive, and mentioned that they will take the learning path specification into account when developing their Education Application Profile. At this state of the project, we envisaged that the learning path specification can be used in a number of ways:

- Educational providers use the specification to describe their offerings and make them available through specific search engines.
- Lifelong learners use the specification to describe their personal learning paths, for instance, to make them available as an example or suggestion to other learners with similar learning goals or to store them in their (TENC) ePortfolio (Berlanga et al., 2008).
- Recommender systems, agents or services can use the learning path specification to support learners in finding a suitable learning path, because it identifies main characteristics to be used in comparing and selecting a learning path.

Requirements for the learning path specification have been derived from a review of literature on curriculum design and lifelong learning as well as observations of current practices that aim to support learner choice (Janssen et al., 2008). The specification is supposed to enable the description of all kinds of learning: formal, non-formal and informal (CEC, 2000). When it comes to selecting a suitable learning path learners are expected to be merely interested in the distinction between accredited (formally recognised) and non-accredited learning. Nevertheless it is important to establish that the specification can be used to describe these three types of learning because they are all particularly important in the realm of lifelong learning. The rest of this section explains the further efforts that have been made to describe in more detail how the learning path applies to non-formal and informal learning (section 2) and to investigate how it could be deployed in the context of a search tool (section 3). Regarding both aspects (description and search) several questions have been formulated at the end of both sections concerning practical deployment. Section 4 describes the case study that has been designed to answer these questions.

4.2 Facilitating description of formal, non-formal and informal learning

According to “A Memorandum on Lifelong Learning”, issued by the Commission of the European Communities, formal learning is learning that occurs in education and training institutions, which leads to recognised diplomas and qualifications (CEC, 2000). The Memorandum further distinguishes between informal and non-formal learning: informal learning is described as “a natural accompaniment to everyday life” (p.8) which is not necessarily intentional learning. Non-formal learning is defined as learning that takes place alongside the
mainstream systems of education and training, for instance at the workplace or in arts or sports, activities which do not necessarily lead to formalised certificates.

Schugurensky (2000) distinguishes formal, non-formal and informal learning characterizing formal education as highly institutionalized; implementing a prescribed curriculum; propaedeutic (each level prepares for the next one); hierarchically organized; and certified. Non-formal education in his view refers to organized educational programs that take place outside the formal school system, whereas informal learning takes place outside the programs offered by formal and non-formal education.

Schugurensky stresses the fact that informal learning can also take place inside formal and non-formal educational institutions when learning occurs independent of the intended goals of the curriculum. Using two categories (intentionality and consciousness) he identifies *three forms of informal learning*:

1. **Self-directed learning (intentional + conscious)**
2. **Incidental learning (unintentional + conscious)**
3. **Socialization (unintentional + unconscious)**

This distinction is relevant to our purpose of testing whether the learning path specification enables description of informal learning as well. However the learning path specification is meant merely to enable description of informal learning with the aim to suggest informal ways to develop competences, drawing from one’s own experiences. This means the learning path specification is only meant to cover conscious informal learning. It would be hard to describe this type of learning in terms of learning activities, because it is often an implicit, gradual process (e.g. learn to live frugally). However, defining unconscious learning as “out of focus” doesn’t solve this problem entirely, because it also exists in relation to a specific type of informal learning the learning path specification is meant to cover: workplace learning.

Whereas formal learning is seen in terms of ‘theory’ (knowledge) and ‘practice’ (application of theory), workplace learning can be viewed as “seamless know how” (Hager, 1998, p. 526). Moreover, this “know how” often is implicit, tacit knowledge. Workplace learning and other informal learning have no formal curriculum or prescribed outcomes. In informal learning, learning outcomes are much less predictable. For our purposes, the lack of predictability does not need to be an issue though, because learners will only describe this learning retrospectively, meaning that, in order to suggest or advice a learning path to others who want to achieve the same outcomes, they will start from what they’ve learned and describe in hindsight how they reached those learning outcomes. Still it should not be underestimated that this may require considerable efforts and reflection on the part of the learner in identifying activities that did or didn’t after all contribute to achieving those outcomes. This is also illustrated by the wide range of terms used as synonyms or examples of learning in studies concerning informal workplace learning: problem solving, feedback, planning, applying, trying things out etc. (Boud & Middleton, 2003; Hoekstra, Beijaard, Brekelmans, & Korthagen, 2006). In terms of the learning path specification these terms can be considered as descriptions of activities that may have lead to certain learning outcomes. Using the learning path specification, with its emphasis on learning activities in relation to learning outcomes, as a template for describing workplace learning could prove to be helpful in identifying activities that did or didn’t contribute to achieving certain outcomes.

Concerning the distinction between different types of learning a major literature search carried out by Colley et al. suggests that there is no clear agreement on the difference between informal and non-formal learning: the terms are used interchangeably (Colley, Hodkinson, & Malcom, 2003). Moreover their study leads them to conclude that it is not possible to distinguish informal
and non-formal learning on the one hand from formal learning on the other hand, in ways that have broad applicability or agreement. Based on these findings the authors suggest it is more sensible to see attributes of informality and formality as present in all learning situations. These attributes of informality and formality can be grouped into four clusters relating to four aspects of learning:

1. **Process**: informality and formality attributes of the learning process can be described for instance in terms of who’s in control of the process (teacher controlled versus student led), whether it involves assessment and what kind of assessment (formative or summative)?
2. **Location/setting**: where does the learning take place (e.g. in an educational institution, at the workplace, etc.) and does it involve certification?
3. **Purposes**: is learning intended or does it happen unintentionally; are learning outcomes determined by the learner or designed to meet needs which are externally determined?
4. **Content**: does the learning focus on acquisition of established knowledge or development of knowledge from experience?

The learning path specification enables identification of a number of these attributes. Attributes relating to the process aspect of learning included in the specification are the metadata elements ‘guidance’ and ‘assessment’. The location/setting aspect is covered by the metadata ‘recognition’, ‘delivery mode’, and ‘teaching place’. Regarding the purpose aspect we stress once more that the learning path specification is meant to describe unintentional learning as well, but this will always be in hindsight and thus relating to learning outcomes which in the process become learning activities (i.e. intentional). Attributes of formality and informality relating to the purpose and content aspect of learning seem hardly relevant when it comes to describing learning paths; learning activities are inevitably intentional and defined. Whether they are achieved through ‘formal knowledge acquisition’ or through ‘learning by doing’ may be of interest to the learner, but we expect pragmatic considerations to prevail in the process of learners selecting a learning path that meets their needs. Still, we have to check this assumption and therefore there are two questions we want to investigate relating to the description of learning paths:

1. Does the learning path specification enable the description of (attributes of) formal, non-formal and informal learning?
2. Do lifelong learners consider it feasible and desirable to describe learning paths in the way suggested by the specification?

### 4.3 Facilitating selection of a learning path

Lifelong learners must be offered means to efficiently choose the learning path that best fits their needs. Taking a decision support perspective we distinguish two stages in the decision making process: screening and choice (Beach, 1997; Rundle-Thiele, Shao, & Lye, 2005). **Screening** involves selecting a number of options one wants to take into consideration. It means narrowing down the number of choice options to a number that can be ‘managed’. Time and attention are scarce resources (Goldhaber, 1997; Schwartz, 2004) and learners would rather invest these resources in developing competences than in comparing all kinds of ways to do so. What is needed then is some tool for the learner to select a limited set of learning paths based on additional criteria so that eventually perhaps three or four options remain left to take a closer look at and finally choose one.

Some criteria will be especially relevant to the screening process, because they represent possibly ‘hard’ constraints on the part of the learner: language, costs, start date, study load, recognition, and delivery mode. If the learner has a maximum amount of money to spend and does not master any foreign languages, learning paths which do not fit these criteria must be filtered out in the screening process. Other criteria are less ‘hard’ or well-defined and more likely to be play a role
in the choice stage - the more in depth comparison of a limited number of learning paths. This in depth comparison is likely to involve elements like description, guidance, and additional information that might be provided through a link. There are quite a number of criteria that could be relevant to finding the most suitable learning path and not all criteria are equally relevant to all learners.

As Fasolo, McClelland and Todd point out the problem of choice overload is not just rooted in the number of options presented to decision makers, but also in the number of attributes related to these options (Fasolo, McClelland, & Todd, 2007). In other words: having to choose from a large number of learning paths is one thing, having to compare even a limited number of learning paths might lead to choice overload when a large number of attributes are related to these options. The study of Fasolo, McClelland and Todd shows that “it is possible for consumers to make good choices based on one or two attributes, when attributes are positively related or consumers care unequally about attributes and choose on the basis of the most important ones.” (p. 23).

To the extent that learners care unequally about the learning path attributes included in the learning path specification this result suggests that progressive disclosure of functionality could contribute to help the learner focus on those criteria that are most relevant for him (Turbek, 2008). Progressive disclosure is “a strategy for managing information complexity in which only necessary or requested information is displayed at any given time.” (Lidwell, Holden, & Butler, 2003).

Regarding the specification’s purpose of facilitating selection of a learning path the following research questions are investigated in addition to the two research questions formulated in section two:

3. Are there attributes of learning, which are relevant for learners, to be considered in deciding on a learning path? If so, which are not identified by the specification?
4. Do learners care unequally about attributes related to learning paths? If learners do care unequally about these attributes, how do they rate the importance of different attributes?

The last two questions are meant to shed light on the issues raised by the research findings described in this section and should lead to recommendations on applying progressive disclosure design in tools supporting search and selection of a learning path.

### 4.4 Case study design

To test the research questions a case study will be carried out involving the most recent formal, non-formal and informal learning paths of approximately 10 lifelong learners. Learners will be selected using an information-oriented selection strategy, more particularly maximum variation of cases (Flyvbjerg, 2006). This strategy is described in more detail below.

For the description of learning paths by the respondents a low-fidelity prototype (mock-up) will be used (Petrelli, 2008) as shown in Figure 1. Respondents will be taken through four successive steps:

The first step involves identification of respondents’ most recent instances of formal, informal and non-formal learning paths. To this end respondents will be introduced with these different ways of learning in lay terms, using examples of how one can learn to use a particular software, or to speak a foreign language, in formal, non-formal and informal situations.

The second step is directed towards validating the learning path specification in two respects:

1. Is the learning path specification suitable to describe formal, non-formal and informal learning alike?
2. Is there a match between the learning path characteristics learners consider relevant in choosing and comparing learning paths and those included in the specification?
Respondents are asked to describe their most recent learning paths by filling in a form, and to think out loud while doing so. The form contains all elements of the learning path specification (Figure 20). The interviewer will make notes in case characteristics or values are unclear, and will focus on identifying learning path characteristics mentioned by the respondent which are not included in the form. These will be noted down on cards to be used in the third step.

![Figure 20: Form to describe learning paths (case study purposes)](image)

Besides, respondents will be asked to describe how they came to choose for each of the three selected learning activities. Did they compare different options? And if so which options did they compare and on which characteristics?

The third step is directed towards establishing the relative importance of all learning path characteristics described by the specification or mentioned in addition by the respondent during the previous steps. Learners will be asked to sort cards containing these characteristics according to their relative importance in selecting suitable learning paths (no importance, some importance, high importance).

In the final, fourth stage of the study learners will be asked to reflect on their experience filling in the form to describe learning paths (see Figure 1): how easy/difficult did they find it to describe learning paths this way; would they want to do it again in order to provide others with examples, would they appreciate to search examples of learning paths described by others in this way?

Using an information-informed selection strategy, respondents will be selected to maximize variation of cases along three characteristics that are likely to influence learning experiences: age, employment status, and sex. In order to ensure variation in the sample we aimed to include male and female respondents in different age groups and with different employment status (unemployed, employed, running a company, retired). We asked colleagues in our department to suggest people in their social network who have varied learning experiences, who are not professionally involved in education and training, and who might be willing to take part in our study.
Results will be used to adapt the first version of the learning path specification when necessary and to derive guidelines for future search and presentation of learning paths.

4.5 References


5 Conclusions and Future Work

In this deliverable we have presented the graphical planning tool for Personal Development Programs, the Hybrid Personalizer, which positions the available learning activities in the graphical overview – based on their relevance for the learner, constraints between the learning activities, the order in which learners plan their program and the learner’s individual preferences, and evaluation work on the learning path description. Together they form the constituent parts of an interactive, personalized environment that lifelong learners can use for planning their learning activities.

Whereas the design of the system is based on extensive background research (reported in D7.1 and in this deliverable), user studies are needed to evaluate the benefits of our approach. For this, we plan a qualitative user study, to be carried out at the UPF at the start of the next study year. A mix of approaches, including questionnaires and thinking-aloud protocols, will be carried out to verify that the interactive planning tool fits and complement current (mainly offline) practices in planning study activities. Usability will be another focus of the study. The main aim of the study is to solicit feedback for future iterations and to support our claims on the benefits of this approach.

A next iteration of the learning path description is planned, with subsequent validation and analysis of strength and weaknesses, including expert feedback.

In DIP-3 we will focus on the design and development of a complementary – or rather prerequisite – tool for the graphical PDP planner. Thus far, we have assumed that the learner goals have already been fixed. However, the learner goal is currently not provided by the PCM. For this reason, we will fill this gap and concentrate on the Competence Matching for Job Opportunities Usage Profile (see DIP-3). The competence matching tool will allow learners to browse and compare various job opportunities and to indicate their ambitions – in a similar, graphical manner as provided by the PDP planning tool.
Appendix 1 – Learning Path Specification

For reference purposes, in this appendix the Learning Path Specification, as presented in D7.1, is displayed in this appendix.
### Elements and attributes of the Learning Path model

<table>
<thead>
<tr>
<th>Name</th>
<th>LD Information model Name</th>
<th>Description</th>
<th>Req!^</th>
<th>Mult</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LearningPath</td>
<td>learning-design</td>
<td>This element specifies the learning path. It describes the actions a learner has to perform in order to attain a competence or competence profile.</td>
<td>M</td>
<td>1</td>
<td>sequence</td>
</tr>
<tr>
<td>identifier</td>
<td>identifier</td>
<td>A unique (locally defined) identifier used to refer to the learning path.</td>
<td>O-&gt;M</td>
<td>1</td>
<td>ID</td>
</tr>
<tr>
<td>title</td>
<td>title</td>
<td>Title of the learning path equals the title of the action when the learning path consists of a single action.</td>
<td>O-&gt;M</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td>uri</td>
<td>uri</td>
<td>Specifies a world-wide unique URI to ??</td>
<td>M</td>
<td>1</td>
<td>anyURI</td>
</tr>
<tr>
<td>learning-objectives</td>
<td>learning activities</td>
<td>Describes the intended outcome for learners by referring to (proficiency levels of) competences and competence profiles (either human readable descriptions or machine readable specifications). A competence profile describes the set of competences a person has to master in order to perform adequately in a particular job or function. Competence is defined as the ability of an actor to act effectively and efficiently in an ecological niche (e.g. occupation, hobby, sport etc.). Competence profiles and competences can have one or more proficiency levels, i.e. levels of mastery (novice, expert, etc.).</td>
<td>O-&gt;M</td>
<td>0..1</td>
<td>sequence</td>
</tr>
<tr>
<td>prerequisites</td>
<td>prerequisites</td>
<td>Describes the entry-requirements for learners in terms of competences and competence profiles (knowledge, skills, and attitudes). (See under: learning-objectives for definitions of competence and competence profile.)</td>
<td>O</td>
<td>0..1</td>
<td>sequence</td>
</tr>
<tr>
<td>components</td>
<td>components</td>
<td>Specifies the building blocks used in the method section</td>
<td>M</td>
<td>1</td>
<td>sequence</td>
</tr>
<tr>
<td>roles</td>
<td>roles</td>
<td>A learning path always relates to the role of a learner. Using IMS-LD as a learning path specification would imply that this element is fixed to one role e.g. 'learner'</td>
<td>M</td>
<td>1</td>
<td>Fixed to 1 role of learner</td>
</tr>
<tr>
<td>min-persons</td>
<td>min-persons</td>
<td>Specifies the number of persons bound to the role before starting the run. See also the metadata element 'startconditions'. When this element is used it should also be reflected in the startconditions.</td>
<td>O</td>
<td>1</td>
<td>nonNegative Integer</td>
</tr>
<tr>
<td>activities</td>
<td>activities</td>
<td>This element contains a choice for different activity definitions, including activity-structure. NB: activities at the level of the Learning Path specification always involve learning activities. If for instance the learning activity consists of being a tutor for a while – in the context of the Learning Path specification it will have to be defined in terms of a learning activity with certain learning activities, rather than a support activity.</td>
<td>O-&gt;M</td>
<td>1</td>
<td>choice</td>
</tr>
</tbody>
</table>

^ This column (Rqd=Required) specifies whether the elements/attributes are considered mandatory (M) or optional (O). “O -> M” indicates that in IMS-LD the element/attribute is optional and we propose to change this to Mandatory for the Learning path specification.
### Name | LD Information model Name | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>learning activity (without activity description)</td>
<td>Any activity performed with the aim to develop a competence. Actions have the same attributes as learning paths: see below identifier, title, version, learning-objectives, prerequisites. The actual activity-description element contains the more or less detailed description of the learning activity(s) to be performed and is not part of the Learning Path specification. To the extent that learners are to be provided with information beyond the learning activities, prerequisites etc. this will be done through the LOM metadata element description.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>A unique identifier used to refer to the action</td>
</tr>
<tr>
<td>title</td>
<td>Title of the action.</td>
</tr>
<tr>
<td>learning-objectives</td>
<td>learning activities Describe the intended outcome for learners by referring to (proficiency levels of) competences and competence profiles (either human readable descriptions or machine readable specifications). A competence profile describes the set of competences a person has to master in order to perform adequately in a particular job or function. Competence is defined as the ability of an actor to act effectively and efficiently in an ecological niche (e.g. occupation, hobby, sport etc.). Competence profiles and competences can have one or more proficiency levels, i.e. levels of mastery (novice, expert, etc.).</td>
</tr>
<tr>
<td>prerequisites</td>
<td>prerequisites Describe the entry-requirements for learners in terms of competences and competence profiles (knowledge, skills, and attitudes). (See under: learning-objectives for definitions of competence and competence profile.)</td>
</tr>
<tr>
<td>complete action</td>
<td>complete-activity Contains a choice of elements to specify when an activity is completed, e.g. when a property has been set, by time-limit, by user choice.</td>
</tr>
<tr>
<td>metadata</td>
<td>metadata Placeholder for metadata about action</td>
</tr>
<tr>
<td>Cluster</td>
<td>activity-structure A cluster is used to group actions (and/or clusters and/or learning paths) that are somehow related, for instance because they compose a set a learner can choose from, or because they have to be studied in a particular order. See below.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>A unique identifier used to refer to the cluster</td>
</tr>
<tr>
<td>title</td>
<td>A header for the grouping of actions, clusters, and/or learning paths.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoundedChoice</td>
<td>AS: structure-type = selection Bounded choice describes a cluster of actions, clusters and/or learning paths a learner has to select a specified number from to complete.</td>
</tr>
<tr>
<td>minimumNumber</td>
<td>number-to-select Specifies the number of elements from the given set that the learner has to minimally select.</td>
</tr>
</tbody>
</table>

M: mandatory, O: optional, Mult: multiplicity, Type: data type
### LD Information model

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Reqd</th>
<th>Multi</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>complete.</td>
<td>M</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>AS: unspecified</td>
<td>O</td>
<td>sequence</td>
<td></td>
</tr>
<tr>
<td>ordered</td>
<td>structure-type</td>
<td>O</td>
<td>1 token (?)</td>
<td></td>
</tr>
<tr>
<td>action-href</td>
<td>learning-activity-ref</td>
<td>M</td>
<td>empty</td>
<td></td>
</tr>
<tr>
<td>cluster-href</td>
<td>activity-structure-ref</td>
<td>M</td>
<td>empty</td>
<td></td>
</tr>
<tr>
<td>learning path-href</td>
<td>unit-of-learning-href</td>
<td>M</td>
<td>empty</td>
<td></td>
</tr>
</tbody>
</table>

### Structure

<table>
<thead>
<tr>
<th>Rolepart</th>
<th>Description</th>
<th>Reqd</th>
<th>Multi</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>The structure defines the ‘work/learning flow’ of a learning path and its constituent parts. In IMS LD this is expressed through the element Method, which consists of one or more play(s) and a statement for the completion of the unit of learning</td>
<td>M</td>
<td>sequence</td>
<td></td>
</tr>
<tr>
<td>Play/act/rolepart/ role-ref = learner</td>
<td>The structure is defined by linking roles to actions, clusters of actions or learning paths by referring to them. At the level of the Learning Path specification the role can be fixed to “learner”. This role will be linked to actions/activity structures which on their part might specify a broader variety of roles.</td>
<td>M</td>
<td>empty</td>
<td></td>
</tr>
<tr>
<td>action-href</td>
<td>learning-activity-ref</td>
<td>M</td>
<td>empty</td>
<td></td>
</tr>
<tr>
<td>cluster-href</td>
<td>activity-structure-ref</td>
<td>M</td>
<td>empty</td>
<td></td>
</tr>
<tr>
<td>learning path-href</td>
<td>unit-of-learning-href</td>
<td>M</td>
<td>empty</td>
<td></td>
</tr>
<tr>
<td>complete learning path</td>
<td>complete-unit-of-learning</td>
<td>O</td>
<td>0..1 choice</td>
<td></td>
</tr>
</tbody>
</table>

### Rules

<table>
<thead>
<tr>
<th>Rules</th>
<th>Description</th>
<th>Reqd</th>
<th>Multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>conditions</td>
<td>Rules can be used to specify whether some actions, clusters or learning paths should be included or excluded under certain conditions.</td>
<td>O-&gt;M</td>
<td></td>
</tr>
</tbody>
</table>

### Metadata

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Description</th>
<th>Reqd</th>
<th>Multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>metadata about</td>
<td>Placeholder for metadata about Learning Path</td>
<td>O-&gt;M</td>
<td></td>
</tr>
</tbody>
</table>

---

**TENCompe]nence – IST-2005-027087**

47/49
### Additional metadata elements of the Learning Path model

(Characteristics of the learning path which are relevant to learner’s screening and eventual choice of a learning path)

<table>
<thead>
<tr>
<th>Metadata</th>
<th>IEEE LOM equivalent</th>
<th>Level (LP / Action)</th>
<th>Description</th>
<th>Reqd</th>
<th>Number</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>language</td>
<td>1.3 Language</td>
<td>LP</td>
<td>Specifies which language(s) the learner needs to know to follow the learning path.</td>
<td>M</td>
<td>10</td>
<td>Language code</td>
</tr>
<tr>
<td></td>
<td>5.11 Language</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>1.4 Description</td>
<td>LP</td>
<td>General description of the learning path.</td>
<td>M</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td>5.10 Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>2.1 Version</td>
<td>LP/A</td>
<td>A version number. Versioning will be necessary to allow for updates of learning paths and enable identification of specific versions.</td>
<td>M</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td>recognition</td>
<td></td>
<td>LP</td>
<td>This attribute only states whether completion of the learning path leads to a formal recognition (diploma/certificate). (N.B.: this is not the same as distinguishing between formal, non-formal, and informal learning. Formal learning not necessarily results in formal recognition).</td>
<td>M</td>
<td>1</td>
<td>Boolean</td>
</tr>
<tr>
<td>startConditions</td>
<td></td>
<td>LP</td>
<td>Several entry or start requirements may hold apart from the required competences (prerequisites) e.g. a specific diploma or course certificate, a minimum age or minimum average grade. Other conditions might relate to practical or pedagogical issues: a minimum number of enrolments.</td>
<td>O</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td>technicalRequirements</td>
<td>4.4 Requirement</td>
<td>LP</td>
<td>Specifies technical equipment and tools a learner needs in order to take this path.</td>
<td>O</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td>4.6 Other platform requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>workload</td>
<td>5.9 Typical Learning Time</td>
<td>LP/A</td>
<td>The total workload in hours.</td>
<td>M</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>assessment</td>
<td></td>
<td>LP/A</td>
<td>Describes which formative and/or summative assessments are in place to determine to what extent the learner has attained the competence.</td>
<td>O</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td>deliveryMode</td>
<td></td>
<td>LP</td>
<td>Describes the modes used for delivery of the learning path, e.g. distance learning, face-to-face teaching etc. We expect this attribute to be important for initial selection (screening) of relevant learning paths to choose from. The vocabulary might be simply: 1. Distance teaching 2. Face-To-Face 3. Mixed model (Distance teaching with virtual meetings / summer schools etc.)</td>
<td>M</td>
<td>1⁶</td>
<td>Vocabulary</td>
</tr>
</tbody>
</table>

³ Maximum number of times the element can occur
⁶ This implies that two separate learning path descriptions will be needed to describe a path which is delivered in two different ways but is otherwise exactly the same.
<table>
<thead>
<tr>
<th>Metadata</th>
<th>Level (LP / Action)</th>
<th>Description</th>
<th>Reqd</th>
<th>Number</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>teachingPlace</td>
<td>LP/A</td>
<td>In case a learning path requires face-to-face meetings the learner needs to know where they take place in order to decide whether this suits him/her.</td>
<td>O</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td>startDate</td>
<td>LP/A</td>
<td>In case there are fixed starting dates for a learning path, for instance in a semester schedule, this information is needed to see whether it fits the learner’s needs and schedule. This attribute will be empty in case learners are free to start whenever they want.</td>
<td>O</td>
<td>10</td>
<td>date</td>
</tr>
<tr>
<td>endDate</td>
<td>LP/A</td>
<td>See startDate. (If completion of a learning path involves time limits this might be expressed through this element e.g. 'six months after start date').</td>
<td>O</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>contactHours</td>
<td>LP/A</td>
<td>Contact hours informs on the hours the learner is expected to attend (virtual) meetings. Teaching place, workload, start date and end date together still don’t suffice to provide the learner with complete picture of the flexibility of the learning path in terms of time, place and pace.</td>
<td>O</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td>guidance</td>
<td>LP</td>
<td>Describes what support is available to learners taking the learning path (tutoring, counselling, helpdesk…).</td>
<td>O</td>
<td>1</td>
<td>string</td>
</tr>
<tr>
<td>costs</td>
<td>LP</td>
<td>Specifies total costs for enrolment and additional expenses (books, tools, etc.). Also specify currency!</td>
<td>M</td>
<td>1</td>
<td>nonNegative integer</td>
</tr>
<tr>
<td>url</td>
<td>LP</td>
<td>Link to a webpage containing more detailed information on the owner of the learning path (person or institution), enrolment, accreditation regulations, facilities for special needs students, contact information etc.</td>
<td>O</td>
<td>1</td>
<td>URL</td>
</tr>
<tr>
<td>contact</td>
<td>LP</td>
<td>Contact information (e-mail address / telephone number)</td>
<td>O</td>
<td>1</td>
<td>string</td>
</tr>
</tbody>
</table>