



Project no. 027087

TENCompetence

Building the European Network for Lifelong Competence Development

Project acronym: Integrated Project TENCompetence

Thematic Priority: 2.4.10

D7.4 Evaluation and Outlook WP7 Results, aggregates internal deliverables ID7.17-ID7.18

Due date of deliverable: 31-10-2009

Actual submission date: 06-01-2010

Start date of project: 01-12-2005

Duration: 4 years

Gottfried Wilhelm Leibniz Universität Hannover

Version 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Project Deliverable Report

D7.4 - Evaluation and Outlook WP7 Results, aggregates internal deliverables ID7.17-ID7.18

Work package	WP7 – Competence Development Programmes		
Task	Task 7.2-7.4		
Date of delivery	Contractual: 31-10-2009	Actual: 06-01-2010	
Code name	D7.4	Version: 1.0	Draft <input type="checkbox"/> Final <input checked="" type="checkbox"/>
Type of deliverable	Report		
Security (distribution level)	Public		
Contributors	Eelco Herder, Philipp Kärger, Adriana Berlanga, Hendrik Drachsler, José Janssen, Marco Kalz, Stef Heyenrath		
Authors (Partner)	OUNL, LOGICA, UHANN		
Contact Person	Eelco Herder (UHANN)		
WP/Task responsible	Eelco Herder (UHANN)		
EC Project Officer	Martin Májek		
Abstract (for dissemination)	Summary of the results of TENCompetence WP7 between M42 and M48. Update of the Learning Path Specification, new version of the Learning Path Editor and evaluation of the Competence Matching Tool		
Keywords List	Learning Path Description, Learning Path Editor, Competence Matching Tool		

Table of Contents

1	INTRODUCTION AND EXECUTIVE SUMMARY	2
2	THE LEARNING PATH EDITOR AS A CHALLENGE FOR THE LEARNING PATH SPECIFICATION	3
2.1	INTRODUCTION	3
2.2	THE LEARNING PATH EDITOR	3
2.3	BUILDING THE LEARNING PATH EDITOR AND CHANGES TO THE LEARNING PATH SPECIFICATION	9
2.4	CONCLUSIONS	10
3	EVALUATION OF THE COMPETENCE MATCHING TOOL	11
3.1	INTRODUCTION	11
3.2	EVALUATION OF THE COMPETENCE MATCHING TOOL	12
3.2.1	<i>Method</i>	12
3.2.2	<i>Procedure</i>	12
3.2.3	<i>Materials</i>	13
3.3	RESULTS	15
3.3.1	<i>Eye-tracking results</i>	15
3.3.2	<i>Desirability results</i>	19
3.4	DISCUSSION	22
3.5	CONCLUSIONS	23
4	CONCLUSIONS AND OUTLOOK	24
5	REFERENCES	26

1 Introduction and Executive Summary

This deliverable is a follow-up of the previous deliverable, D7.3¹, in which we presented the Learning Path Specification and its associated Learning Path Editor, the Graphical Planning Tool and the Competence Matching Tool.

In Chapter 2 we discuss the latest version of the Learning Path Editor (LPE), which has been integrated into the TENCompetence infrastructure. The development activities for the LPE were also conceived as an evaluation of the pragmatic quality of the underlying model, the Learning Path Specification.

In Chapter 3 we present the results of the user evaluation of the Competence Matching Tool. A qualitative, case-based user study, making use of eye-tracking technology and a desirability toolkit. This study provides various insights in how users search for job vacancies and how interactive visualizations can support this search.

Finally, in Chapter 4, we summarize the results of WP7 during the lifetime of the project. We discuss the concepts and tools that have been developed and the insights gained from these activities. Further, we provide an outlook for these concepts and tools – in particular those that have been integrated into the TENCompetence Infrastructure.

¹ <http://dspace.ou.nl/handle/1820/1979>

2 The Learning Path Editor as a Challenge for the Learning Path Specification

2.1 Introduction

In Deliverable 7.3 we described the Learning Path Specification and the tool that was being developed at the time to enable description of learning paths in accordance with the specification, the Learning Path Editor. In this chapter we describe the current version of the Learning Path Editor and discuss how the development contributed to the evaluation of the Learning Path Specification.

The framework used for the evaluation of the learning path specification is based on theories and research in the area of conceptual model quality (Beck, 2002; Krogstie, 1998; Moody, 2005; Nelson, Poels, Genero, & Piattini, 2005) and distinguishes three aspects of model quality:

1. *Semantic quality*: does the model/specification represent essential features?
2. *Syntactic quality*: does the model/specification express what it intends to express in a correct way, i.e. in accordance with the syntax rules of the modelling language?
3. *Pragmatic quality*: is the model/specification easily comprehended and used by the stakeholders for its intended purpose?

The *immediate* purpose of the learning path specification is to guide system development, with the eventual aim to describe learning paths in a formal and uniform way. In this sense the process of developing the Learning Path Editor can be considered as an evaluation of the pragmatic quality of the learning path specification. Pragmatic quality concerns the question whether the specification is easy to understand and easy to use. Pragmatic quality of a model, specification or system is high when it is:

- a. Unambiguous (i.e. concepts and relations have a clear single meaning);
- b. Internally consistent (i.e. the model does not contain contradictions);
- c. General (i.e. concepts are as independent as possible from any specific application or domain).

Assessing pragmatic quality is not merely a matter of asking users how they evaluate the specification regarding these criteria. Though software developers will understand these criteria and are able to identify a specification's flaws related to these criteria at face value (i.e. merely by reading the specification), more flaws are likely to come to light in the process of building a tool that draws on the specification.

The chapter is structured as follows. In the next section we give an overview of the functionalities of the Learning Path Editor. We continue with a discussion on how the process of developing the Learning Path Editor evoked a number of changes to the learning path specification and end the chapter with some conclusions.

2.2 The Learning Path Editor

The Learning Path Editor (LPE)² consists of three different *views* that correspond to different tasks in connection with the description of learning paths:

1. Handling of learning paths, i.e. keep an overview, choose to change existing learning paths or to create a new learning path (Master view)
2. Describing the characteristics of a learning path (Metadata view)
3. Designing a learning path (Design view).

The metadata view and the design view reflect the learning path specification.

² Available as part of the TenCompetence infrastructure: <http://dspace.ou.nl/handle/1820/2182>

The Learning Path Editor opens in the Master view (Fig. 1), which provides an overview of all learning paths created within the Liferay Community, in which the LPE is integrated. The learning paths are listed alphabetically and a date indicates when the learning path was last changed.

When the user clicks on the *collapse* and *expand* buttons, additional information becomes visible, provided that the author of the learning path gave this information when describing the learning path: description, delivery mode, and workload. Further control buttons that are provided for each learning path include Copy, Delete, Metadata, and Design.

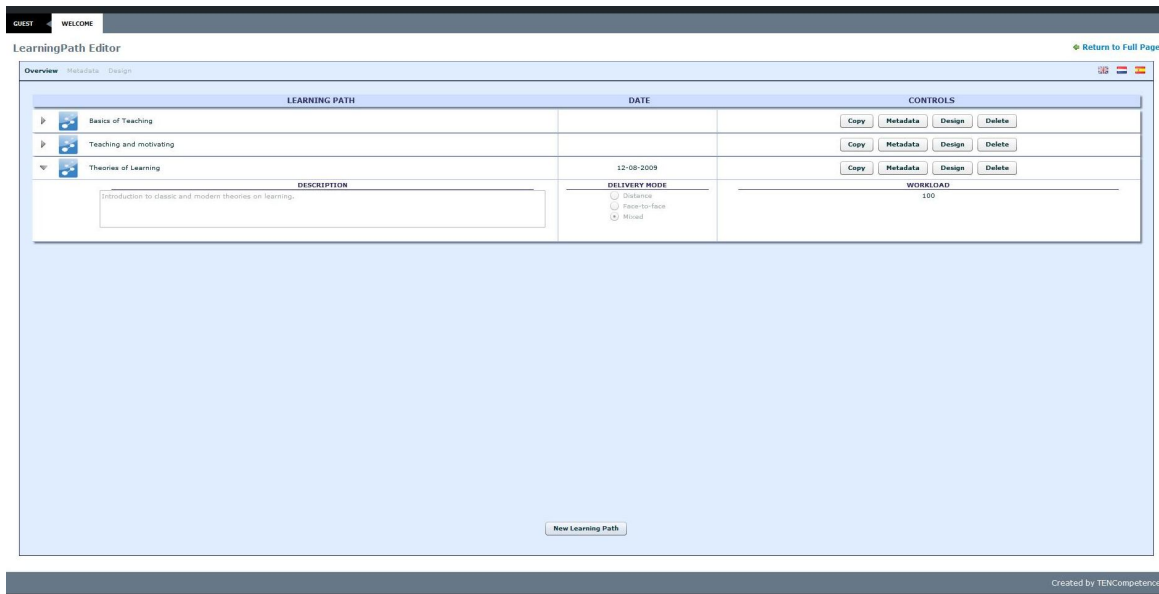


Fig. 1. Learning Path Editor Master view

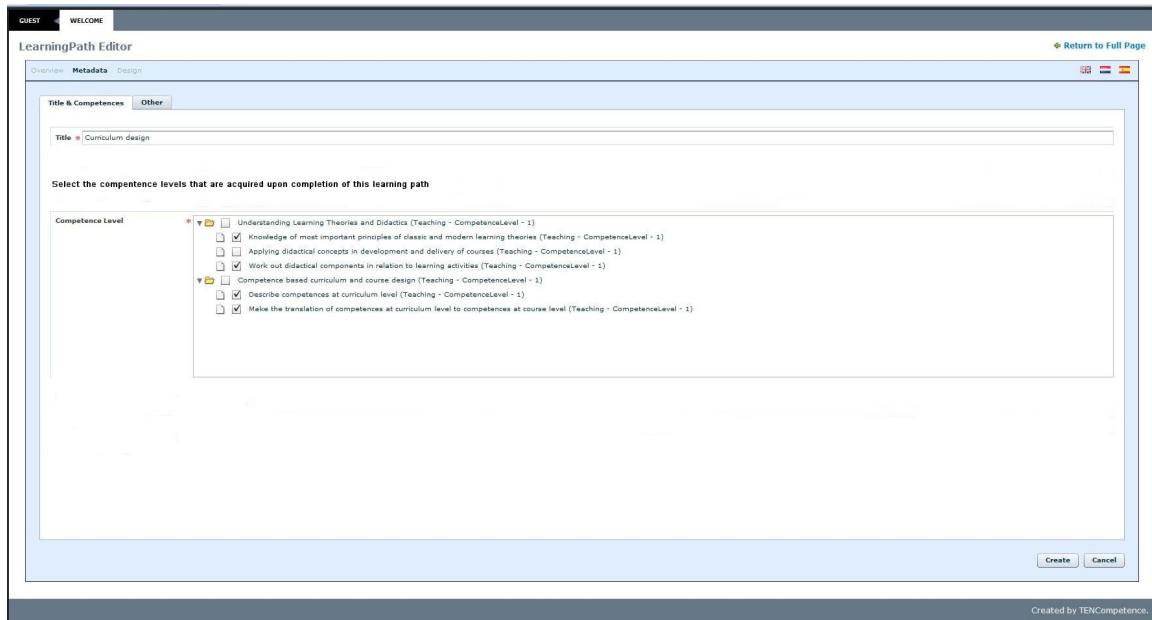
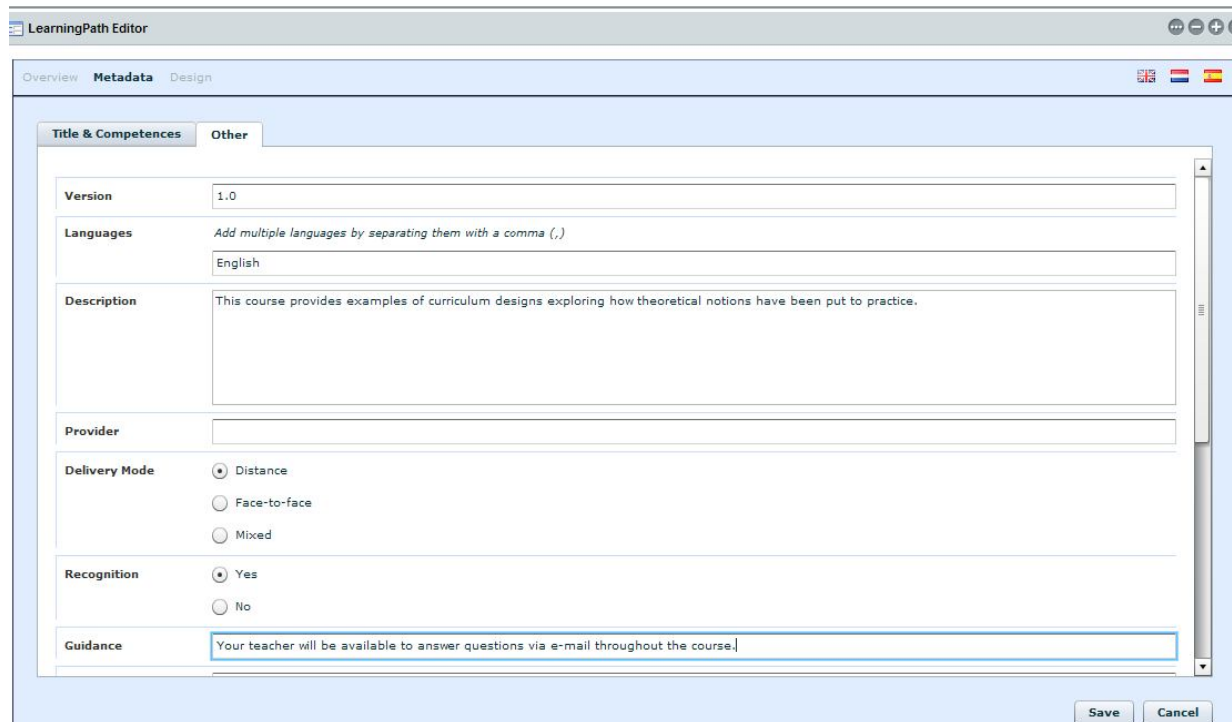


Fig. 2. Learning Path Editor Metadata view (Title & Competences)

By clicking the button *New Learning Path* the metadata view is activated (Fig. 2). In this view the author enters a title and short description of the learning path, and specifies the competence levels that are attained upon completion of the learning path. The competence levels displayed in the metadata

view are all competence levels that are defined within the Liferay Community that the Editor is deployed in.

Further, the author has the option to specify additional characteristics of the learning path (Fig. 3.) which may be used by learners in the process of searching a suitable learning path, for example language, costs and delivery mode.



The screenshot shows the 'LearningPath Editor' interface with the 'Metadata' tab selected. The 'Other' sub-tab is active, displaying the following fields:

- Version:** 1.0
- Languages:** English (with a note: 'Add multiple languages by separating them with a comma (,)')
- Description:** This course provides examples of curriculum designs exploring how theoretical notions have been put to practice.
- Provider:** (empty text field)
- Delivery Mode:** Distance (selected), Face-to-face, Mixed
- Recognition:** Yes (selected), No
- Guidance:** Your teacher will be available to answer questions via e-mail throughout the course.

Buttons for 'Save' and 'Cancel' are located at the bottom right of the form.

Fig. 3. Learning Path Editor Metadata view (Other metadata)

Once the necessary information has been provided the learning path can be saved and the user will be redirected to the overview, now including the newly created learning path. The actual design of the learning path still has to take place, though. For this, the author clicks the *Design* control button for the learning path. This activates the design view (Fig. 4).

The design view shows the competence levels previously selected in the metadata view, at the left hand side of the screen, as a cue for the author. In the Design view the author can add actions or existing learning paths and group them in clusters to specify particular subsets, such as sequential ordering and choice options. (Fig. 5 to 8). Note that the user has to specify whether the learning path (top level) constitutes a sequence, free order or parallel grouping of learning actions.

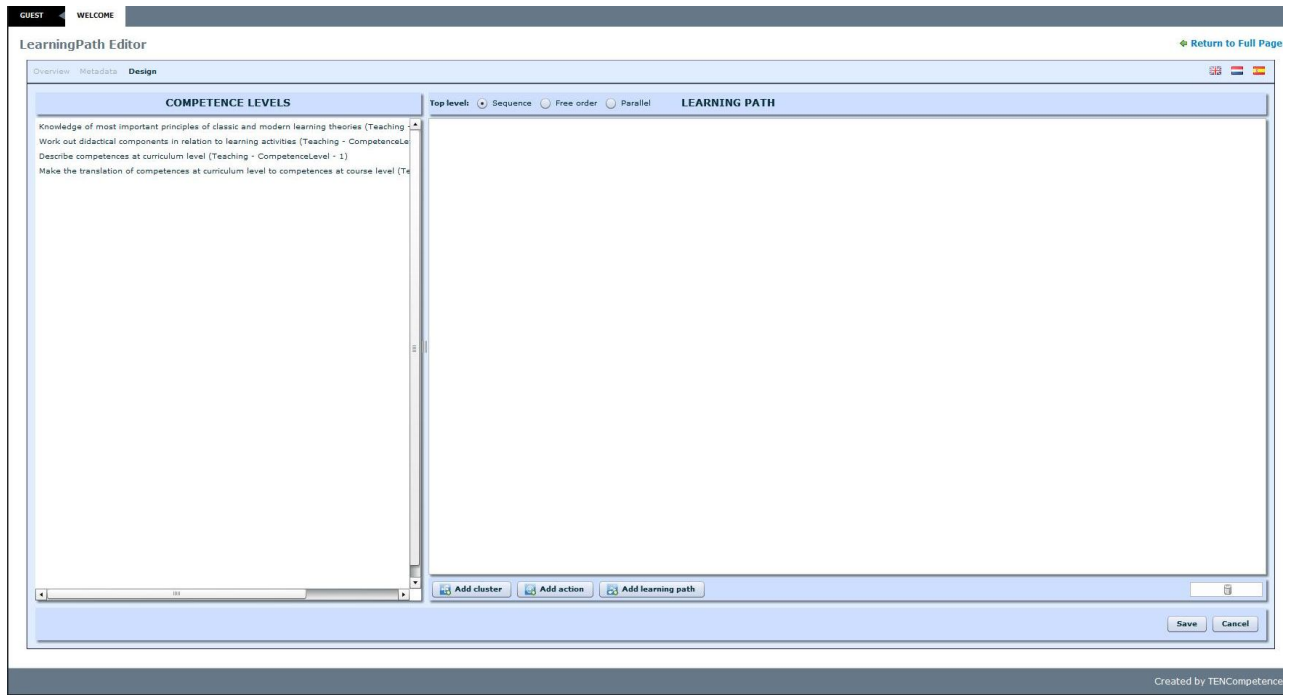


Fig. 4. Learning Path Editor Design view

To add an Action, the author clicks the “Add action” button at the bottom of the right hand pane. Next a dialogue box appears which asks to provide a title and a web address for the action (Fig. 5). This web address may refer to all kinds of actions: a simple instruction to read a book, a complete course, a game or simulation, a test etcetera.

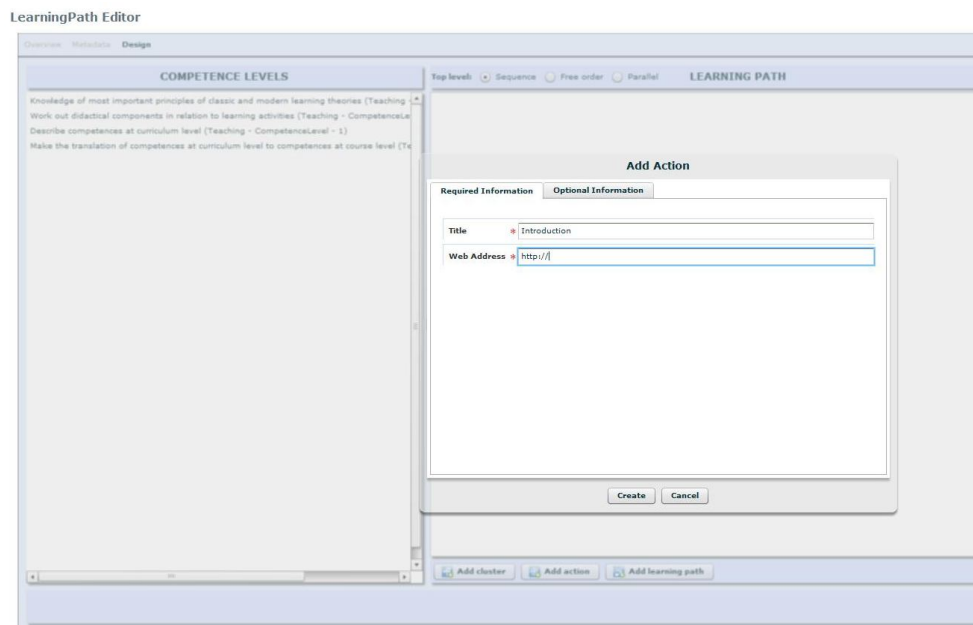


Fig. 5. Learning Path Editor Design view: Add Action

Alternatively, the author can choose to include existing learning paths in the design, for instance to express that a university degree is build up of a bachelor programme and a master programme, each of which has been designed as a separate learning path. An important difference between a learning

action and a learning path is that the latter has an explicit connection with one or more competence levels. Completion of a learning path equals mastery of the competence levels associated with the learning path, whereas completion of a learning action means that ‘some contribution’ to attainment of one or more of the learning path’s competence levels has been achieved. This means that whenever a strict relation needs to be clear between parts of a learning path and one or more competence levels, for instance to adapt a learning path to individual learners’ prior knowledge or learning experiences, these parts must be designed as learning paths within a learning path.

When the author wants to add a learning path, she is presented with an overview of existing learning paths (Fig. 6). Clicking on a learning path in the list will highlight the competence levels the learning path helps to acquire in the list of competence levels next to the list of learning paths, enabling the author to compare them with the list of competence levels in the left hand pane of the main screen.

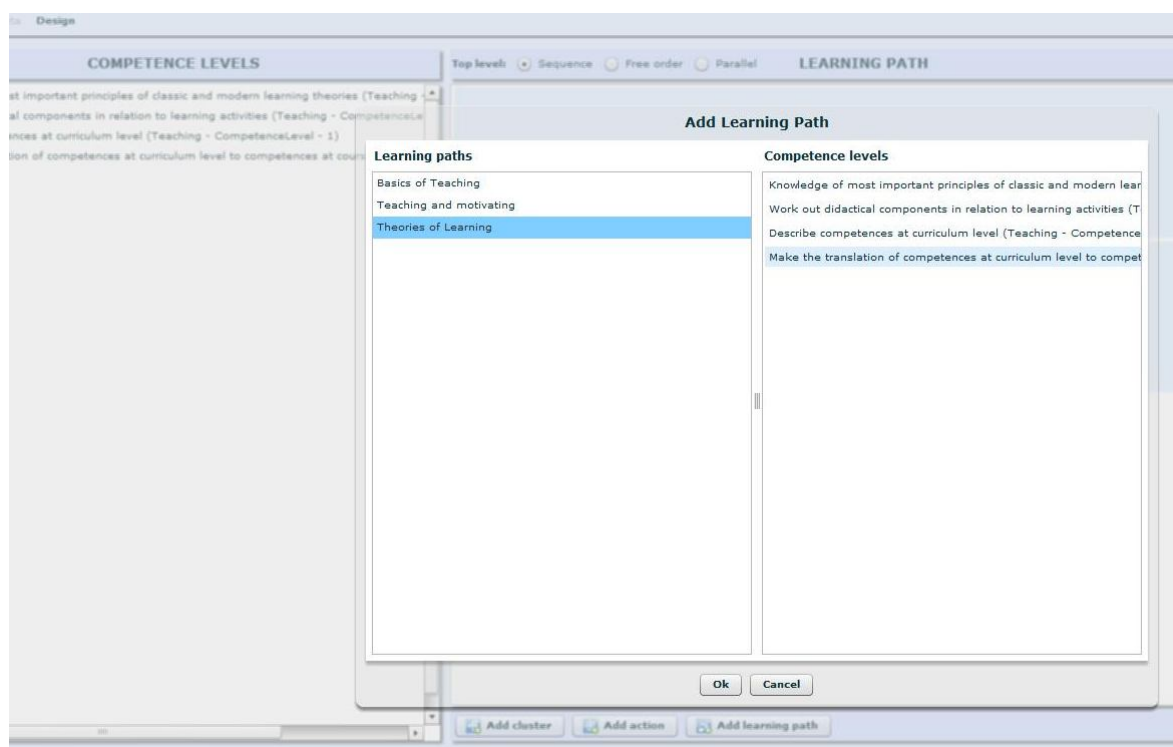


Fig. 6. Learning Path Editor Design view: Add Learning Path

The author has now added three learning actions and a learning path to the learning path that is being designed. However two of these should be studied in parallel, for the sake of synergetic effects. In order to express this, the author has to group the action and, in this case, the added already existing learning path into a cluster of the type ‘parallel’. She first creates the cluster by clicking “Add Cluster” at the bottom of the right hand pane and again a dialogue box appears, in which she can specify what type of cluster she wants to create (Free order, Sequence or Parallel), provide a title for the cluster and, in the case of a Free order or Parallel cluster indicate whether all the actions in the cluster have to be completed or that the learner is free to choose a certain number (Fig. 7.).

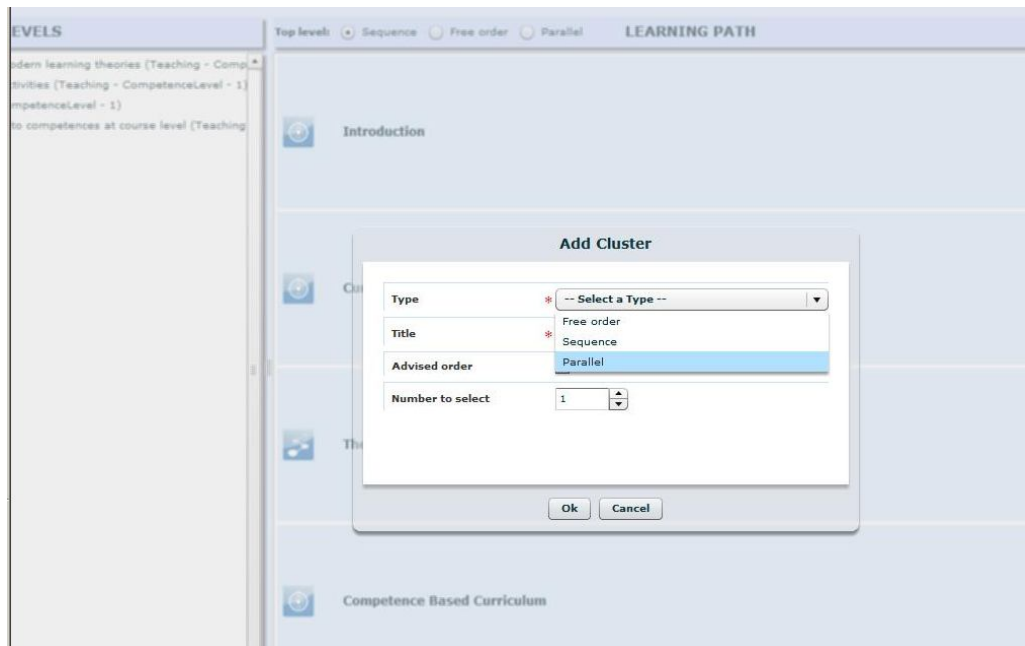


Fig. 7. Learning Path Editor Design view: Add Cluster

Once the cluster has been added the author can drag and drop the required actions and paths to the cluster. Fig. 8 shows the result in the Design view.

This is how gradually the author builds a learning path out of learning actions or existing learning paths. Supported by the list of competence levels selected at the very start of the process of creating a learning path.

Further technical documentation can be found at <http://sourceforge.net/projects/tencompetence/>

The Learning Path Editor code is also available through <http://hdl.handle.net/1820/2182>.

Finally Deliverable 3.5 (and specifically the section reporting on ID3.24 “Design & architecture of the integrated system”) provides information on the Learning Path Editor in relation to the wider system.

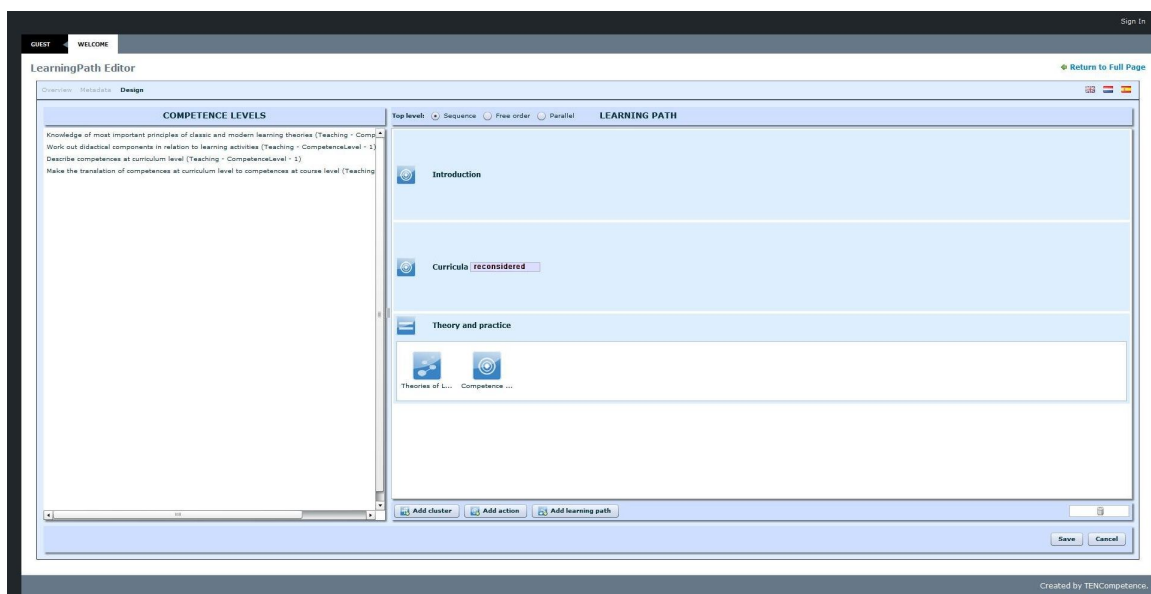


Fig. 8. Learning Path Editor Design view: action and learning path included in parallel cluster

2.3 Building the Learning Path Editor and Changes to the Learning Path Specification

The Learning Path Editor is designed as a tool to be used by experienced authors, to create learning path descriptions following the learning path specification. In the TENCompetence infrastructure these learning paths are then presented to learners through the Personal Development Planner.

Some notable differences between the current Learning Path Editor and the original design are:

1. The original distinction between owned learning paths and co-edited learning paths has not been implemented.
2. There is no connection between learning actions and competence levels. Only learning paths are connected to competence levels in the strict sense that upon completion of the learning path the associated competence levels have been acquired.
3. Following point 2 the design view no longer has a top pane representing competence levels which upon selection highlight associated learning actions in a left hand pane. Instead the left hand pane now shows a list of all competence levels the author has selected for the currently designed learning path to cover.

At the outset of the Learning Path Editor design and development it was decided, for reasons of efficiency, that the user interface already developed for the Graphical Planning Tool would be re-used for the Learning Path Editor. This meant that the Editor would be developed in Flash. However, at some point along the line it became clear that the integration of Flash in the Liferay environment was more complicated than anticipated and in fact constituted a technical problem that could not be solved in a reasonable amount of time. At that point a switch was made from Flash to Flex. This in turn meant that the initial design had to be adapted, not only because of differences between Flash and Flex but also to enable completion of the development within the time limits set.

The switch from Flash to Flex also meant that eventually five software developers have worked in less or more detail with the learning path specification: three developers involved in developing the Learning Path Editor and two involved in developing the Activity Navigator.

The software developers involved in developing the Editor had not been involved in the development of the specification, so that to start with, they had to rely on the information contained in the schema and the information model (Janssen, Hermans, Berlanga, & Koper, 2008). Clarifications were requested on a number of issues that led to several minor as well as some more profound adaptations of the schema (binding).

Minor changes to the learning path specification are those which concerned the translation of the specification into this specific schema rather than the specification itself. The term ‘specification’ is somewhat confusing in this respect, as it is used both for conceptual models (e.g. a UML model) and the technical implementation of these models in a schema using a particular syntax, in our case XML (cf. (Klein, Fensel, Harmelen, & Horrocks, 2000)). As is the case in natural language, the things we want to express can be expressed in many different ways grammatically, representing different nuances perhaps, but still bearing the same message/meaning. Similarly the minor adaptations made to the schema represent ‘grammatical’ or ‘syntactical’ changes:

- In the first version of the schema the attributes (metadata) of LearningPath, LearningAction and LearningActionsCluster were grouped in a container element Metadata. One of the software developers proposed to take the metadata element ID out of the container element, so that the three elements become more directly accessible. Even if we call it a minor change it clearly brought about an important improvement in terms of pragmatic quality.
- For reasons of consistency/clarity the element LearningPathRef was changed into the element ExternalLearningPath and an ID was made mandatory for this element (LearningActions and LearningActionsClusters are also ‘declared’ separately and then referenced internally through an ID).
- The element Learning Actions as container element within the LearningActionsCluster element was removed because it appeared superfluous and therefore confusing. The schema now indicates

that a LearningActionsCluster contains one or more LearningActionsRefs, LearningActionsClusterRefs, or ExternalLearningPathRefs.

Three *more profound changes* were made, two of which were related to the rendering of a learning path in the Activity Navigator:

- The metadata element ‘Title’ which was optional in the first version of the specification was made mandatory to enable proper rendering in Activity Navigator.
- Added an attribute AdvisedOrder to the LearningActionsCluster, which provides information on how the Cluster should be presented to learners. The attribute specifies whether the order in which LearningActions are included in the cluster is the recommended order to study them in. When the Cluster is of the type ‘sequence’ or ‘parallel’ “Yes” means that the learner can diverge from the presented order. “No” means that the order is mandatory. When the type is ‘free order’ the value “Yes” means that the learning actions in the cluster should be presented as a recommended order (though the learner could still diverge from it). “No” means that they can be presented and followed in a random order.
- Removal of the restriction that a LearningActionsCluster contains at least two elements. Though there is some logic to the restriction, e.g. it hardly makes sense to define a group of 1, it is not necessary to specify such a restriction and indeed in the case of the learning path specification it proved a mistake since the Cluster element is also used to specify the ordering overall of LearningPath and it should be possible to create a LearningPath consisting of a single action.

All in all two out of the six changes made, involved an improvement of pragmatic quality of the specification in the sense that these changes made it easier for software developers to read/interpret the specification. The other four changes (ID, Title, Advised order, and Restriction) involved optimising the specification for practical deployment in different tools, most notably the Learning Path Editor. A description of all adaptations has been provided in the Release Change notes published along with the new version of the specification (Janssen et al., 2008).

2.4 Conclusions

Development of the Learning Path Editor took place in two stages, because the initial plan to develop the Editor in Flash had to be abandoned due to integration issues with the Liferay portlet environment. The time constraints related to the switch to Flex also required that the original design of the Learning Path Editor was modified and restricted to represent basics.

The process of developing the Editor has prompted several adaptations of the Learning Path Specification as well, which have contributed to the specifications pragmatic quality in the sense that it has become easier to interpret the specification and to deploy the specification in relevant tools.

3 Evaluation of the Competence Matching Tool

3.1 Introduction

The Competence Matching Tool³ (described in more detail in Deliverable 7.3) provides an interface to search for job offers, making use of a competence based search. This functionality is different from traditional job portals like *Monster* or *jobpilot* (see also Jansen et al, 2005). In contrast to these job portals, the Competence Matching Tool compares competences of a user with the required competences for a certain job. It extends the *normal* job search with the competence dimension and allows the users to see their abilities in the context.

The Competence Matching Tool gives the users the possibility to judge their current position and potentially required competences for the labor market. In a second step, the users can identify competence gaps in their competence profiles which they have to further develop to reach the required competence level of a target job profile.

Besides this innovative job search the Competence Matching Tool also provides an innovative way to view relevant jobs. It ranks relevant job offers according to the suitability to a certain user profile and visualises them on a two-dimensional graph. The vertical axis of the graph represents how close the match of a job with the competence profile of a user is and the horizontal axis represents the match with the search preferences of the user (job location, type of work, salary etc.) see Figure 9.

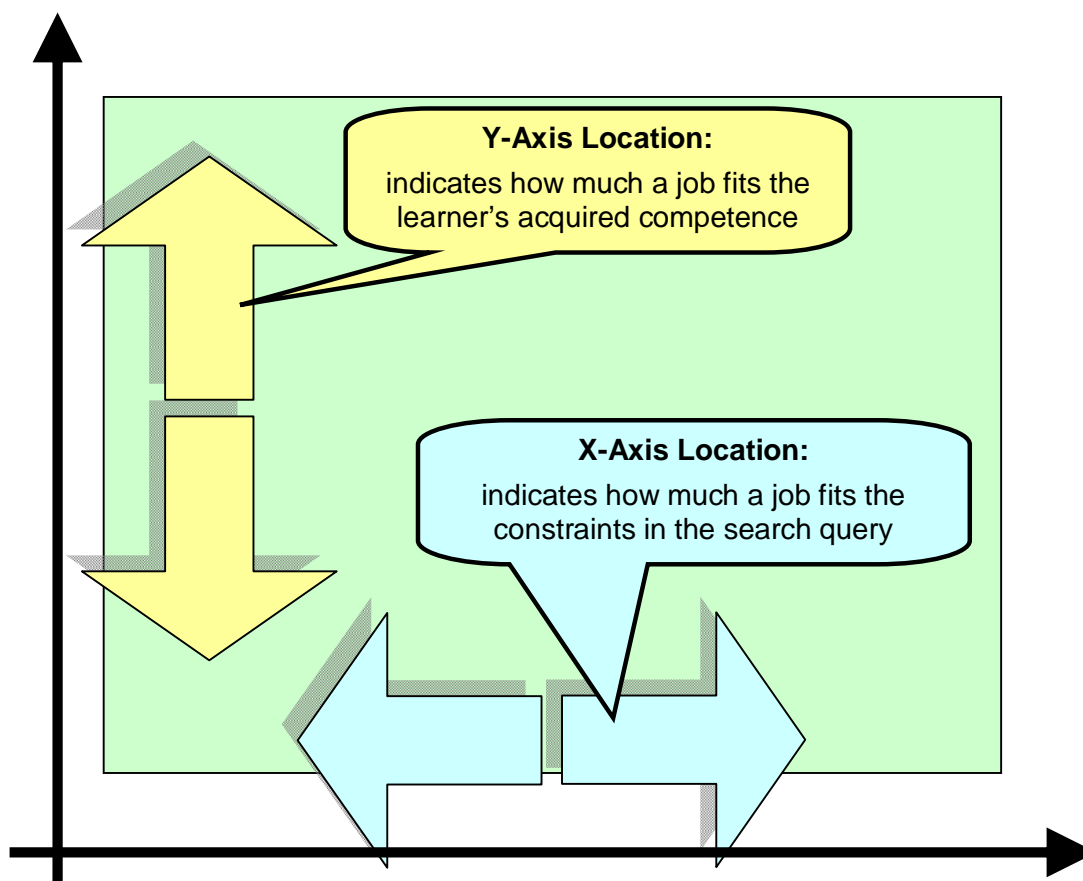


Fig. 9. Placing jobs to visualize competence gaps and user preferences.

- ³ Available at <http://tencompetence.cvs.sourceforge.net/viewvc/tencompetence/wp7/CompetenceMatcher/>

The chapter is structured as follows. In section 3.2 we present the usability evaluation of the Competence Matching Tool by introducing the evaluation approach, the procedure for the evaluation, and the used materials. In section 3.3 we present the results of the evaluation for the different evaluation measures (Eye-tracking results and Desirability results). Finally, we discuss the findings of the evaluation and provide suggestions for the future development and improvement of the Competence Matching Tool.

3.2 Evaluation of the Competence Matching Tool

In order to get a first usability and satisfaction evaluation of the Competence Matching Tool we conducted a qualitative case study (Field and Hole, 2003) with five participants. Qualitative case studies with a limited amount of users have been shown to be effective for these types of evaluation (Nielsen, 2009).

The participants received a prepared user profile and had to find the most suitable jobs for their profile by using the Competence Matching Tool. The main questions for the evaluation were:

1. Do the users see the advantage of the competence based job search and can use it properly?
2. Are the users able to use the job graph view in an efficient manner?
3. Which changes and additions will improve the usability of the Competence Matching Tool?

3.2.1 Method

To test the Competence Matching Tool we applied a combination of two different usability evaluation methods, which address both the actual user behaviour (the eye-tracking method) and the user's attitude (a 'desirability' evaluation) (Nielsen, 2008).

1. For the eye-tracking method we used a *TOBI eye-tracker*⁴ device. This eye-tracker enabled us to monitor user interactions in a quantitative way by recording the eye movements of the users during the runtime. In that way we could ask the participants during the interaction if, why, and where they struggle with the handling of the Competence Matching Tool. In addition, we were able to reflect together with the participants their interactions by looking at the eye-tracker recordings.
2. The second part of the evaluation was conducted on the basis of the *Desirability Toolkit* of the Microsoft Cooperation (Benedek & Miner 2002). We assessed the overall user acceptance and desirability of the system using *Product Reaction Cards*. The participants received 118 cards with descriptive words. The set contains 60% positive and 40% negative/neutral cards. The users were asked to pick the 5 most suitable cards from the 118 available to describe the Competence Matching Tool. Afterwards, we asked for an explanation why the users picked each of the 5 cards. During this information exchange the users offered many valuable information about their interaction with the Competence Matching Tool as well as their thoughts about the approach to use competences to identify most suitable job profiles. Therefore, the Desirability Toolkit offers qualitative information about the usability of the Competence Matching Tool in a standardised form.

3.2.2 Procedure

In total five users with an interest in educational technology and varying backgrounds (computer science, psychology, education) participated in the qualitative study; each of them used the Competence Matching Tool for the first time.

The users received a written and an oral introduction to their task. First they were asked to inspect their (predefined) user profiles in the Competence Matching Tool, to become aware of competences of their identity (Figure 10). Afterwards, we calibrated the eye-tracking device to the current user and

⁴ <http://www.tobii.com/>

he/she started to search for the most suitable jobs. On average the participants used the tool for 40 minutes. Afterwards, we applied an online version of the Desirability Toolkit and asked the users to pick exactly five cards that best describe the Competence Matching Tool. This interview situation took in average 25 minutes.

Competence Profiles		
Competence Profiles	Competence	Aquired Level
Language Skills	German	5
	French	3
Baking Skills	Dough preparation	3
	Baking bread	3
	Baking pastry	5
	Product decoration	5
Sales Skills	Advertising	5
	Customer Advice	5
Management Skills	Reporting	2
	Work Planning	3
	Conflict Management	2
	Accounting	3
	Leadership	2
	Budgeting	2
Cooking Skills	Prepare Meals	4
	Create Recipes	4
	Plan Menus	4
	Check Quality Ingredients	4
	Follow Strict Time Schedule	2

Fig. 10. Summary of the user profile description in the Competence Matching Tool.

3.2.3 Materials

As mentioned in the introduction, the Competence Matching Tool provides two innovative procedures to search for most suitable jobs: the competence based search to find suitable jobs (Figure 11) and the ranking and visualisation of the most suitable jobs on a two-dimensional graph, where the vertical axis represents how close the match of an advertisement is with the user's competence profile and the horizontal axis represents the match with the preferences of the users (Figure 12). We focused the usability evaluation on these two aspects and report the results in the following sections.

Competence Matching Service - Job Search

Job Profile

Location Country: City:

Category *Multiple categories possible.*
 Professional Services
 QA & Quality Control
 Real Estate
 Restaurant & Food Service
 Retail
 Sales

Type of Job *Multiple types possible.*
 Full Time
 Part Time
 Temporary

Occupation *Multiple occupations possible.*
 Research
 Sales
 Food Services
 Management

Salary Minimum Euro, maximum Euro.
Similar for any other preferences, such as Industry and Function.

Competences

Please indicate to what extent the job needs to match your competence profiles

My competences
 are not considered during search.
 are roughly similar to the results.
 ' levels are the same or higher as in the results.
 ' levels are the same as in the results.

Fig. 11. The Job search form of the Competence Matching Tool.

Competence Matching Service - Job Results

[Refine your search](#) [Start new search](#)

Matches that fit both your competences and preferences

List Plot

Language Skills
 Baking Skills
 Sales Skills
 Management Skills

Overview

Baker
 that prepare and se...
 prep and s...
 quality service in all areas to meet our standards in I...

Pastry Cook
 Pastrycooks mix, bake, fill, ice and decorate all kinds of cakes and pastries. Pastrycooks may specialise as chocolatiers, patissiers, ice carvers, sculptors...

Pastry Cook

Type	Full Time
Category	Restaurant & Food Service
Date	2009-05-01
Institute	Deutsches B�ckerst�bchen

Pastrycooks mix, bake, fill, ice and decorate all kinds of cakes and pastries. Pastrycooks may specialise as chocolatiers, patissiers, ice carvers, sculptors and festive cake decorators. Pastrycooks stand most of the day. They may work broken

salary
 city
 country
 category
 type

Fig. 12. The relational graph view of available jobs in the Competence Matching Tool.

3.3 Results

3.3.1 Eye-tracking results

The eye-tracking analysis was very fruitful regarding suggestions for the improvement of the Competence Matching Tool. Here we report on the two most important aspects of this eye-tracking analysis. We discuss a *heatmap* and a *gazeplot* representation of the competence profile of the user, the job search form, the list view of the suitable jobs, and the relational graph view.

The heatmap shows an overlay of all heatmaps of the 5 users that participated in the eye-tracking evaluation. Color in a heatmap represents the level of focused interest or invested time of users. Here the green colour represents a lower interest / invested time while the red colour represents a high level of interest / invested time.

The gazeplot represents the movement of the eyes from one point of interest in the Competence Matching Tool to another one. It shows the connections between these points of interest and the spend time in seconds. Equally to the heatmap representation the gazeplot is also an overlay of the eye-movements of all participants in the evaluation. Each of them is represented with a different colour.

3.3.1.1 Competence profile of a user

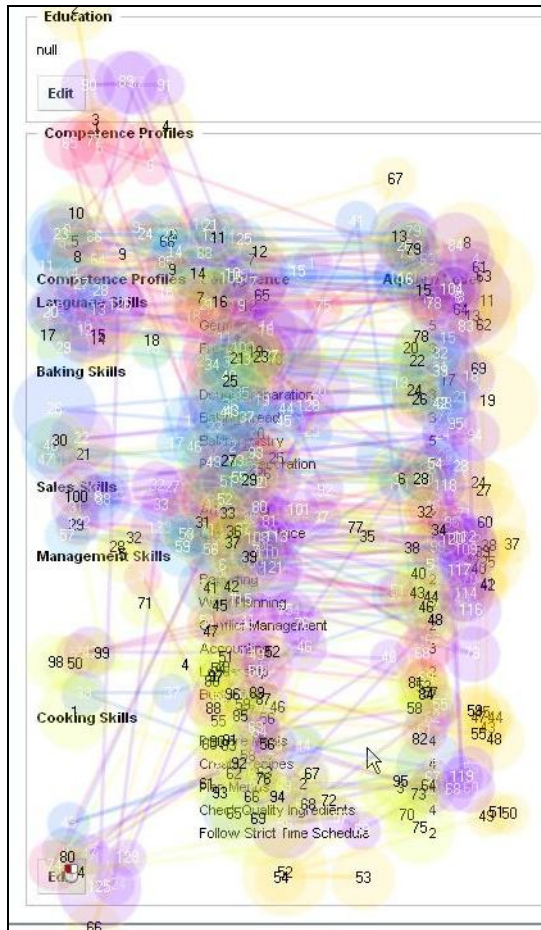


Fig. 13. Gazeplot of the Competence profile of a user.

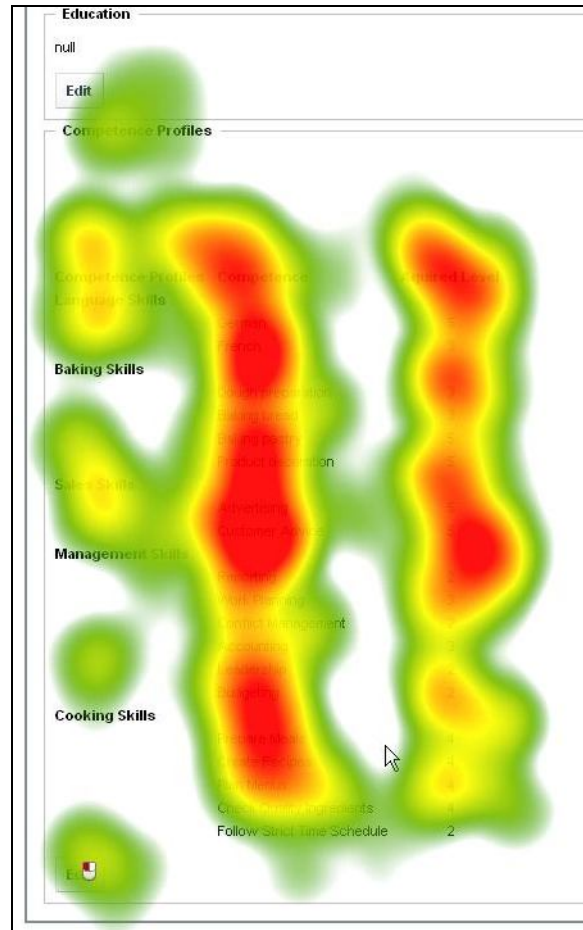


Fig. 14. Heatmap of the Competence profile of a user.

Figure 13 and 14 show that the participants carefully looked at the upper information of the competence profile of the user, but they spend less time to read the lower information of the competence profile, especially the competence values on the lower right corner. This was already

obvious during the evaluation phase, so we asked the participants why they spent less time on this area. All of them reported that they had difficulties to relate a certain competence to the related competence level. The main reason for low attention was that it was hard to read and took a lot of effort, so they tended to skip that section. Another explanation for this phenomenon is given by Nielsen (2006): users tend to concentrate mainly on the top-left area ‘above the fold’.

3.3.1.2 Search form

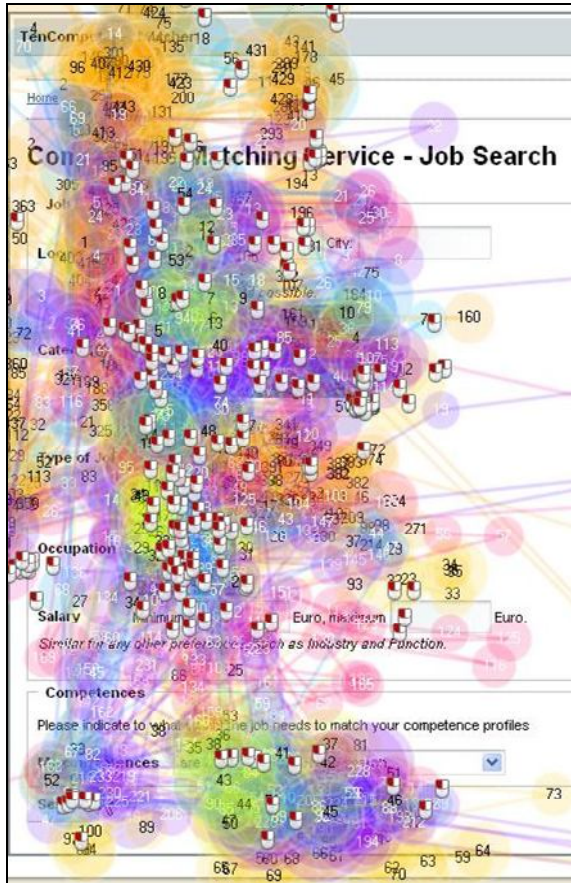


Fig. 15. Gazeplot of the job search form.

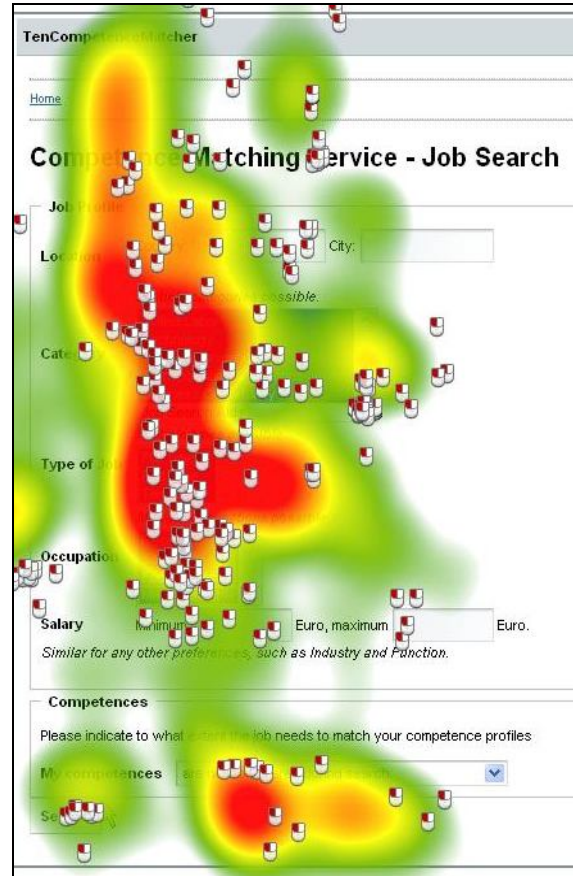


Fig. 16. Heatmap of the job search form.

Figure 15 and 16 show that the participants actively selected different search options within the job search form. Via the eye-tracking approach no problems were identified: the user’s attention was at the areas where it should be for the form.

However, the desirability evaluation later on made clear that the participants had problems to decide what they should select in the *Competence combo-box* as almost all mentioned options were meaningless to them. Further, it was hard to understand for the participants that the competences of the user profile have an effect on the search results while they are not shown in the search form.

3.3.1.3 Job list view

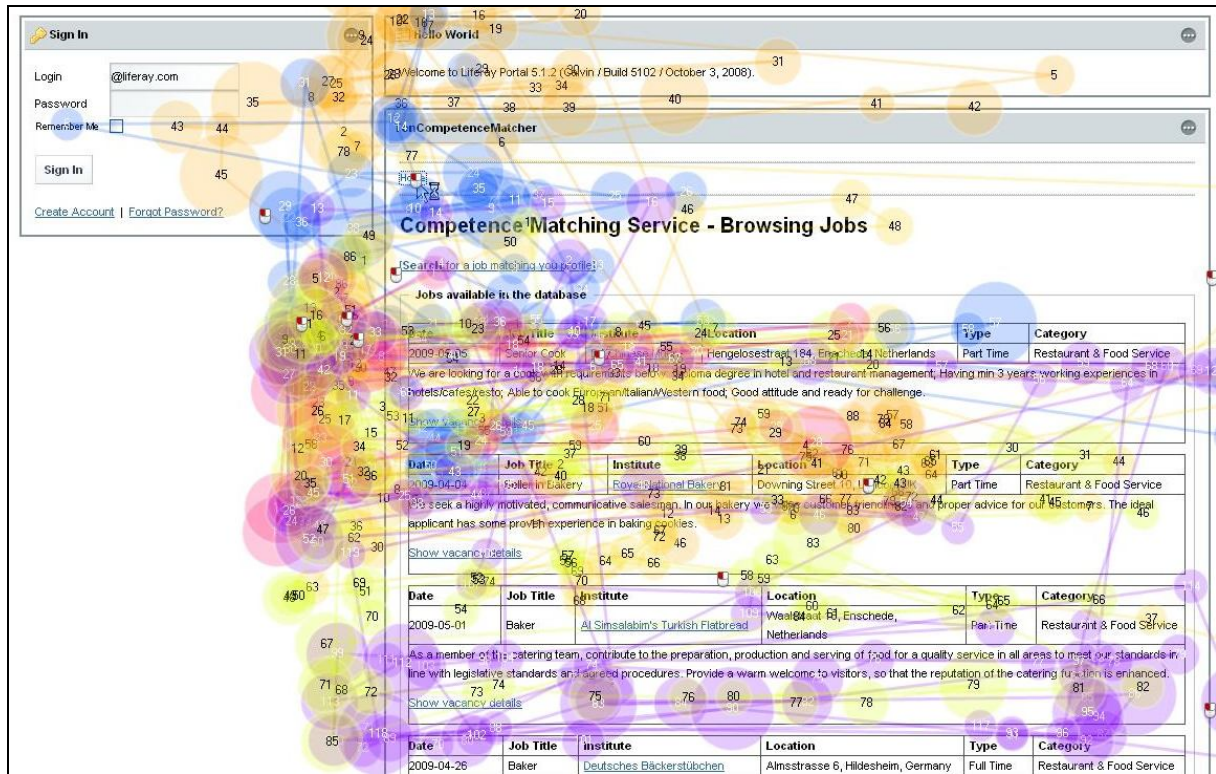


Fig. 17. Gazeplot of the job list view.



Fig. 18. Heatmap of the job list view.

Figure 17 and 18 show that most of the participants only looked at the top-3 ranked jobs in the list view and spent significantly less time on viewing the lower jobs presented. The gazeplot in Figure 17

shows how the different participants read through the job offers and that they had a different job description in focus. No major problems were found on this screen.

3.3.1.4 Job results view

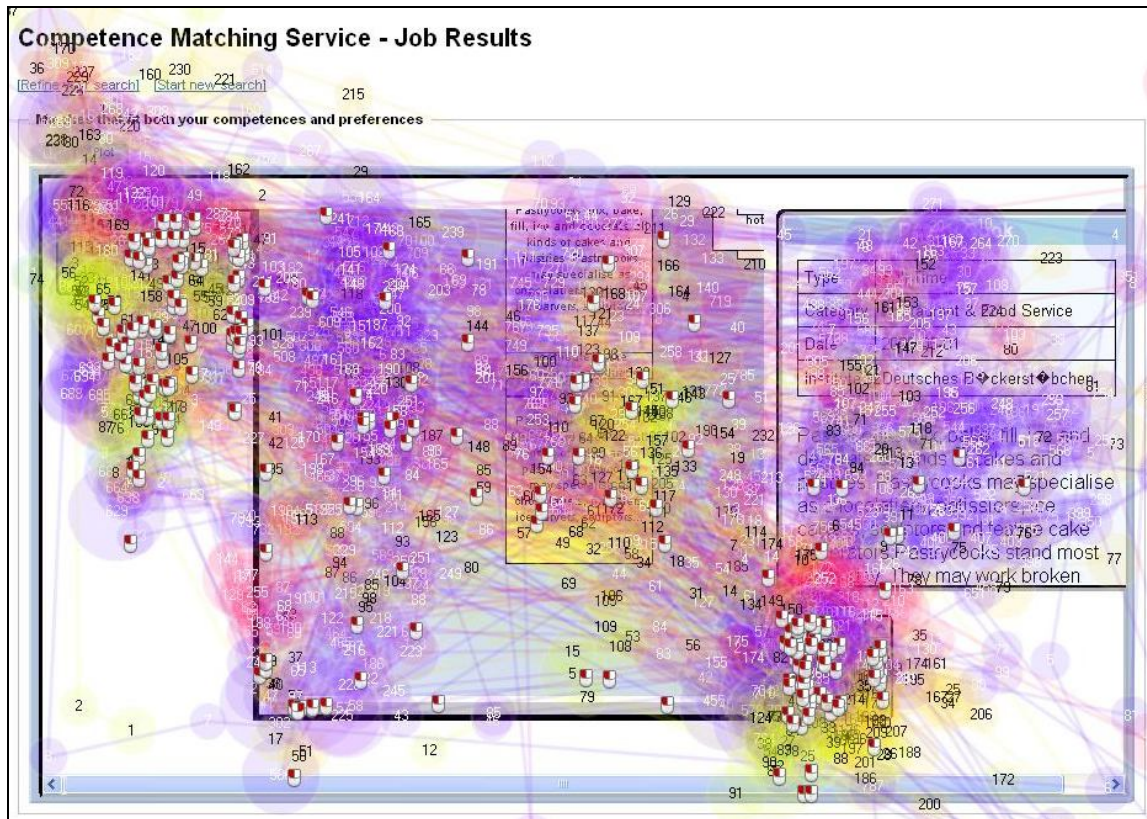


Fig. 19. Gazeplot of the relational graph view.

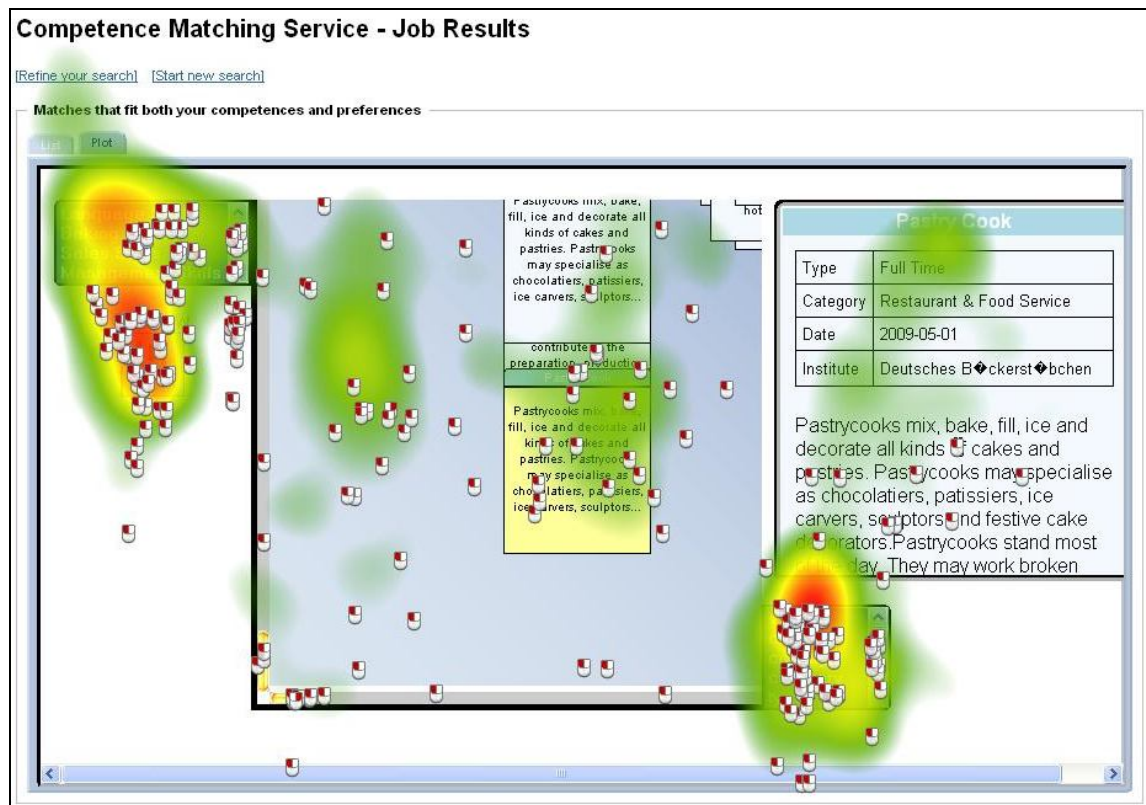


Fig. 20. Heatmap of the relational graph view.

Figure 19 and 20 show that there are two hotspots in the graph interface. These hotspots are on the one hand on the overview window and on the other hand on the categories shown from the search.

The interpretation of these results can be that the users explored the influence of these two “navigation tools” on the graph view. On the other hand it can be expected that this exploration phase should stop after some time. But the eye-tracking analysis reveals that there is a constant attention to these aspects of the interface, which could also mean that they are distracting from the more important graph view. Surprisingly, the participants almost never used the scroll bars of the graph to navigate through the offered jobs.

The gazeplot in Figure 18 confirms these findings and shows also the order of attention. Often users followed the sequence “Overview window – graph – categories” while most attention was spent on the overview window and the categories in order to discover the most suitable jobs. Further, the gazeplot in Figure 19 also shows that the participants watched at the jobs in the graph view and the detailed job description but that had no effect on the heatmap. This is reasonable, as the jobs were located differently and moved on the graph depending on the combination of preferences by the participants.

3.3.2 Desirability results

We categorized the results of the Desirability Toolkit into three main objectives: *User Interface*, *Technology*, and *Innovation*. Figure 21 shows the online Version of the Desirability toolkit that was used for the evaluation of the Competence Matching Tool. In Table 3.1 we present the selected cards and the comments of the participants.



Fig. 21: The online version of the Desirability Toolkit created by the CELSTEC institute. 118 reaction cards are shown to the participants and they should select at 6 out of them.

Table 3.1: Reaction cards and interview comments.

Feedback	Reaction cards	Description
<p>User Interface</p>	<p>Competence based search</p> <ul style="list-style-type: none"> • 2 x Difficult • 3 x Confusing • Hard to Use 	<p>Competence based search</p> <ul style="list-style-type: none"> • I don't understand how my competences in the search form are involved in the results. The selection statements were just meaningless to me. • I find it very difficult to find the most suitable job based on my competence. That made the job search much more complex. Normally, I just look for something suitable and then I call the employer. • The results of my search and the competence related search aspects were unclear to me. • Most of the time the job descriptions are overlapping each other. I would expect that they show really different locations based on my preferences, which would make it clearer. • I did not understand how I can influence the search criteria. The jobs were moving from one corner to another but I did not know where the most suitable jobs are located. • I find the technical solution very old-fashioned, even when the search is based on competences. It is not supportive or transparent what is happening behind the system.

Feedback	Reaction cards	Description
	Job graph view <ul style="list-style-type: none"> • Annoying • Unpredictable • Inconsistent 	Job graph view <ul style="list-style-type: none"> • The interface was not very appealing and inconsistent. • There are no labels at the graph; a user does not know which position represents the most suitable job. • I still can not predict how the graph view reacts based on the search criteria. I just don't get how this tool supports my job search. It really needs some more indicators and support features to make it easier and more intuitive.
Technology	<ul style="list-style-type: none"> • 2 x Dated • 2 x Gets in the way • 2 x Too Technical 	<ul style="list-style-type: none"> • 90% of the application is a simple search engine. • I always received an error message when I tried to view the plot. That also shows that the tool is really in the prototype status and has to improve a lot. • The error messages were annoying. • The categories do not sound anything to a user. The descriptions have to be much more clearer. • Why do I see this error message all the time?
Innovation	<ul style="list-style-type: none"> • Personal • Innovative • Unconventional • Simplistic • Easy to use • Stable • Predictable • Integrated 	<ul style="list-style-type: none"> • Search based on my personal competences makes it more personal. • The idea of visualizing jobs and distances to an individual profile is an innovative idea even when the technical solution is still a bit difficult. • The idea to visualise jobs on a graph to show their relation to an individual profile is an innovative idea even when the technical solution is still a bit prototypical. • It's just a search engine. • Besides some error messages it is easy to use. • I always got the same results. • Not much surprises here. • It seems to be integrated in other systems.

3.3.2.1 User Interface

Regarding the category *User Interface* we received 3 times the reaction card 'Confusing', 2 times 'Difficult', and once 'Hard to use'. The participants criticized during the interview that they did not understand how the competences of their profile were involved in the job search. The selection of jobs was unclear to them and there was no explanation or help text available. Additionally, the participants found the graph view 'Hard to use', 'Inconsistent', and even 'Annoying'. They got frustrated because the graph view builds upon the competences in the user profile and the search results and was therefore even less reasonable for them.

In a similar fashion as in the user profile and the job search form, the users missed information about the graph view and how to use it properly. They did not understand when a job was suitable, because the graph offered no indicators for that. They also missed help information for using the graph view and for manipulating the results.

3.3.2.2 Technology

Regarding the category *Technology*, the participants selected two times the reaction card 'Dated', 2 times 'Gets in the way', and two times 'Too Technical'. Parts of these comments are related to a technical problem on one of the evaluation sessions. There was sometimes an error message on the screen mentioning that no job profiles could be placed in the graph view.

In addition, the participants criticized that most of the text descriptions in the Competence Matching Tool were too technical and meaningless for them.

3.3.2.3 Innovation

Regarding the category *Innovation*, the participants selected a broad range of reaction cards. None of the cards were selected twice. Surprisingly, they selected some oppositional cards compared to the cards they selected earlier. Also in their comments they valued the competence based search as an interesting, innovative, and personal way of searching what stands in contrast to their earlier expressions. The general concept seems to be a reasonable and interesting approach for the participants. One of the participants literally said:

“The idea to visualise jobs on a graph to show their relation to an individual profile is an innovative idea even when the technical solution is still a bit prototypical. “

3.4 Discussion

In this section we provide answers to the initial evaluation questions to point out critical aspects and potential improvements.

Question 1: *Do the users see the advantage of the competence based job search and use it properly?*

We can say that overall, the participants found the competences based search approach interesting and innovative. Most of them believed that this can help to offer more personalised search results. Regarding the second part of the question (*Can the participants use the competence based search properly?*) we have to admit that the participants were confused and not able to use the tool in the most efficient manner. They missed supportive information how to affect the results of the Competence Matching Tool. That can be solved by improving the user interface and adding additional help information.

Question 2: *Are the users able to use the job graph view in an efficient manner?*

The graph view got many criticisms by the participants and most of them found the graph view very difficult to use. One reason for that might be that people always stick to traditional information models like ‘lists’ or ‘table overviews’. They need more time to get used to new information models and especially to discover their advantages. However, the participants mentioned some comprehensible improvements like adding a legend to the graph, emphasizing the most suitable job offers, and adding descriptions to the axis of the graph. By implementing these practical hints and by adding supportive help texts, the graph view might be more useful for the end-users.

Question 3: *Which general changes and additions will improve the usability of the Competence Matching Tool?*

Based on the conducted evaluation we can suggest a list with practical improvements to increase the usability and desirability of the Competence Matching Tool. On a general issue is the connection between the competence profile of the user and the job search form. In case of competence-based job search, the search form and the user profile should be locally combined in one page. This especially applies, when the job search form should also work as a gap analysis between the current user profile and the desired job descriptions. More specific recommendations are listed below.

User profile

- The competences and their related competences level should be highlighted with different background colours to improve the readability.

Job search form

- The wordiness of the job search form should be less technical and therefore renamed to the needs of the end-users. For instance, the participants had problems to decide what they should select in the Competence combo-box because almost all options were meaningless to them.

Job graph view

- The presented jobs in the job graph should not overlap each other rather than being aligned otherwise they hardly can be read.
- The graph view needs further descriptions on the axis and some indicators to mark the most suitable jobs.
- The graph view should support at least the most popular web browsers like Firefox, Safari and Internet Explorer to avoid confusing error messages.

3.5 Conclusions

From the evaluation of the Competence Matching Tool it has become clear that the users recognized and appreciated the innovative features of the approach. However, there are still some areas for improvement before it becomes part of the standard distribution of the TENCompetence infrastructure.

Apart from the suggestions for improvement of the user interface, an interesting observation is that whereas users recognize the benefits of graphical interfaces, they still prefer list-based results, which are easier to interpret: the higher a result is in the list, the more relevant the result is. This ease of interpretation may yield for simple keyword-based searches, but it is definitely not the case for tasks in which many, orthogonal facets are important – facets that at the start may not be apparent to the user. As discussed in Deliverable 7.3, a job search depends on many different factors: suitability of a job in terms of function description, experience level and industry; education and background required for the job; more mundane factors such as location and salary play a significant role as well. This wide range of factors is taken into account by the matching algorithms, but the screen real estate and the users' difficulties in dealing with multi-dimensional visualizations hinders the effective presentation of these various dimensions simultaneously. For this reason, graph-based views can be used as an effective interface, but classical list-based views should be the default option.

Another observation is that users do not have a clear mental model of the concept of competences, which is an obvious obstacle for competence-based search and exploration of job vacancies. This emphasizes the need for standardized competence models, which provide users a framework to relate to and for understanding the relation between the competence matching and the ranking of search results. In order to reach this point, standardization efforts such as the Dutch Colo project⁵ together with new technologies that show how to put these competence models into effective use play a significant role.

⁵ <http://www.colo.nl/>

4 Conclusions and Outlook

In TENCompetence WP7 we explored new horizons in how lifelong learners could better create and work with *personal development plans* and how human resource managers can support them, making use of appropriate computer support. Extensive background research, prototyping and evaluation has led to a variety of approaches and tools. In this chapter we highlight the most significant results of WP7 during the lifetime of the project.

As there are no current practices on computer-supported competence development programs, the work in WP7 has deliberately been exploratory, aiming at proofs of concepts rather than tools that can directly be put in the market. Several concepts and tools, however, have reached a level that is mature enough for integration in the TENCompetence infrastructure. A major focus in WP7 has been on the development of graphical user interfaces and the combination of various techniques for providing lifelong learners, with different needs and backgrounds, with the tools that they need.

The **Learning Path Specification** is one of the first specifications for the modeling of personal development plans or curricula. On the level of learning activities, specifications exist, such as IMS Learning Design. However, these specifications include a number of constructs that the learning path itself does not require and vice versa. The LPS distinguishes itself from related specifications (such as XCRI, CDM and MLO-AD) that focus only on formal learning, because the LPS also enables description of non-formal and informal learning. Like any path, a learning path has a Start (prerequisites) and a Finish (learning goals). Learning actions help learners to get from the start to the finish. In the data model it is described how these actions can be combined with one another and which information on a learning action should be given (for example, title, description, language, workload) in order to make meaningful combinations. The data model is specified in UML with a serialization into XML. End-users interact with the LPS via the Learning Path Editor.

The **Learning Path Editor** is a novel, experimental tool that enables the description of all kinds of personal and professional development, whether consisting of formal, non-formal or informal learning, in accordance with the learning path specification. The LPE is designed to be used by experienced authors (e.g. human resource managers, career counselors, curriculum designers) in order to create competence development programs within their organizations. By dragging and dropping the relevant learning activities, the managers can easily create flexible programs, with required and optional learning activities, with partially specified orders and partially freedom for the learner. Normally, doing so requires quite some time and effort, but the LPE provides a good balance between complexity (more options, but less user-friendly) and usability (less options, but more user-friendly).

The Learning Path Specification has gone through several iterations of refinement, which are reported in earlier deliverables. The main focus of these refinements was finding the right balance between the richness and complexity of related specifications (among others IMS-LD, XCRI) and simplicity. The current specification is a result of various evaluations revisions of the objects, their relations and their metadata, which led to a lean yet well-specified model. The process of developing the Learning Path Editor can be considered as an evaluation of the *pragmatic quality* of the learning path specification. Pragmatic quality concerns the question whether the specification is easy to understand and easy to use. Both the Specification and the Editor have become part of the TENCompetence Infrastructure – which reflects the mature status of this work.

An important aspect of WP7 research has been the **graphical user interface**. Based on solid theories of human-computer interaction and information visualization, various iterations of prototypes of the Graphical Planning Tool, the Learning Path Editor and the Competence Matching Tool have shown how these principles can be applied to competence development programs. Graphical user interfaces have the advantage that they use simple interaction mechanisms such as drag-and-drop, but they require strong visual metaphors, which typically only evolve in time.

The **Graphical Planning Tool** for Personal Development Plans allows learners to plan their upcoming learning activities by dragging and dropping - with instant feedback on their choices. The Planning Tool proposes a first plan, based on the learner's needs and preferences, which is visualized as a

learning path that the user can interact with. The main reason for this interactive approach is that 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 and let the learner interact and make the choices herself.

For lifelong learners, the **Competence Matching Tool (CMT)** aids exploration of the next career opportunities. Similar to common career sites as Monsterboard it allows you to search for jobs on specific criteria, such as location, industry and salary. In addition, it compares all competences that are required for a vacancy with your own competences. An important step in lifelong learning is learning what you want to learn. Career perspectives are a major motivating factor. This is exactly what the CMT shows you.

The main challenge addressed by these tools – a challenge of which the importance has become apparent due to the background research, the development activities and the evaluation of the tools – concerns the many facets that are related to competence development and the matching of job vacancies with a lifelong learner's own competence profile. As discussed in the previous chapter, this requires finding a balance between the users' preference for simple keyword searches with list-based output and the more precise matching that results from faceted search and user interaction with multi-dimensional visualizations of learning opportunities. This does not only yield for lifelong learners: human resource managers face the same challenge when working with the Learning Path Editor. The drag-and-drop mechanisms embedded in the LPE, based on a redesign of the Graphical Planning Tool, show the effectiveness of rich visualization approaches in the field of lifelong learning. It also shows that these visualizations require several iterations before they can effectively be used by lifelong learners.

As discussed before, the many facets that play a role in finding and selecting opportunities for lifelong learning, ask for matching and recommendation algorithms that take these facets into account and that allow for balancing between these – often orthogonal – facets. Users of the Personal Development Planner will get an initial recommendation for a learning path that is provided by a **Hybrid Personalization and Recommendation System**. This system combines four atomic personalization services:

- a *positioning service*, which estimates how relevant a learning activity is for the learner
- a *navigation service*, which advises the next best choices, based on what peer learners have done
- a *preference-based service*, which trades off and reasons about diverse learner preferences (including cost, distance, type of examination, duration, credits) and ranks these options accordingly
- an *algorithmic curriculum planner*, which assures that all prerequisites between learning activities are taken into account (for example, that one should first attend Basic Math before one can attend Advanced Math).

The concepts of **hybrid personalization** (are inspired by the fields of hybrid recommender systems. In adaptive (educational) hypermedia systems, the separation of model, view and controller are well taken into account, but in the actual implementation they often end up being merged. With our implementation we have shown the practical benefits of keeping these layers separated: it allows for flexible recommendations based on various dimensions, which is particularly useful for personal development plans, given the many facets that are important to them: the competences that they provide, the level of knowledge and skills that one learns, the domain in which they are applicable, the nature of courses and examination, availability, schedule, planning, costs.

The Hybrid Personalization and Recommendation System, as integrated into the Personal Development Planner is a simplified version of the original prototype, as developed in WP7. Reasons for simplification include performance issues (the LSA-based positioning service was too slow, for example) and the current lack of metadata, which is mainly due to the absence of standardized competence models. As mentioned in the previous chapter, we currently see various activities – such as the Colo initiative – that fill this gap.

5 References

- Beck, B. (2002). Model evaluation and performance. In A. H. El-Shaarawi & W. W. Piegorisch (Eds.), *Encyclopedia of Environmetrics* (Vol. 3, pp. 1275-1279). Chichester: John Wiley & Sons.
- Benedek, J., Miner, T. (2002). Measuring Usability: New Methods for Evaluating Desirability in a Usability Lab Setting. Presented at UPA conference, July 2002.
- Field, A. and Hole, G (2003). *How to Design and Report Experiments*. SAGE Publications, ISBN 0-7619-7382-6
- Herder, E. and Kärger, P. (2008) Hybrid Personalization for Recommendations. Proc. ABIS 2008
- Jansen, B.J., Jansen, K.J and Spink, A. Using the Web to Look for Work: Implications for Online Job Seeking and Recruiting. *Internet Research* 15 (1), 2005, pp 49-66
- Janssen, J., Hermans, H., Berlanga, A. J., & Koper, R. (2008). *Learning Path Information Model*. <http://hdl.handle.net/1820/1620> .
- Klein, M., Fensel, D., Harmelen, F. v., & Horrocks, I. (2000). The Relation between Ontologies and Schema-Languages: Translating OIL-Specifications to XML-Schema. In *Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conference on Artificial Intelligence ECAI-00*. Berlin, Germany, August 20-25, 2000.
- Krogstie, J. (1998). Integrating the Understanding of Quality in Requirements Specification and Conceptual Modeling. *ACM SIGSOFT Software Engineering Notes*, 23(1), 86-91.
- Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55, 243-276.
- Nelson, H. J., Poels, G., Genero, M., & Piattini, M. (2005). Quality in conceptual modeling: five examples of the state of the art. *Data & Knowledge Engineering*, 55, 237-242.
- Nielsen, J. (2006). Screen Resolution and Page Layout. Alertbox, July 31, 2006. http://www.useit.com/alertbox/screen_resolution.html
- Nielsen, J. (2008) When to Use Which User Experience Research Methods. Alertbox, October 6, 2008. <http://www.useit.com/alertbox/user-research-methods.html>
- Nielsen, J. (2009) Discount Usability: 20 Years. Alertbox, September 14, 2009. <http://www.useit.com/alertbox/discount-usability.html>