

ID 7.13+17 – Validation of learning path description, Implementation of the Learning Path Manager and Editor

Citation for published version (APA):

Herder, E., Kärger, P., Berlanga, A., Janssen, J., & Heyenrath, S. (2010). *ID 7.13+17 – Validation of learning path description, Implementation of the Learning Path Manager and Editor*.

Document status and date:

Published: 06/01/2010

Document Version:

Peer reviewed version

Document license:

CC BY

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

Take down policy

If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 02 Jul. 2022



Internal Project Deliverable Report

ID 7.13+17 – Validation of learning path description, Implementation of the Learning Path Manager and Editor

Work package	WP7 – Competence Development Programs		
Task	Task 7.2-7.3		
Date of delivery	Contractual: 30-11-2009	Actual: 05-01-2010	
Code name	ID7.13	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, José Janssen, Stef Heyenrath		
Authors (Partner)	OUNL, LOGICACMG, UHANN		
Contact Person	Eelco Herder (UHANN)		
WP/Task responsible	Eelco Herder (UHANN)		
EC Project Officer	Martin Májek		
Abstract (for dissemination)	Update of the Learning Path Specification, new version of the Learning Path Editor.		
Keywords List	Learning Path Description, Learning Path Editor,		

Table of Contents

EXECUTIVE SUMMARY	2
1 THE LEARNING PATH EDITOR AS A CHALLENGE FOR THE LEARNING PATH SPECIFICATION	3
1.1 INTRODUCTION	3
1.2 THE LEARNING PATH EDITOR.....	3
1.3 BUILDING THE LEARNING PATH EDITOR AND CHANGES TO THE LEARNING PATH SPECIFICATION	9
1.4 CONCLUSIONS.....	10
REFERENCES	11

Executive Summary

In this internal deliverable 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.

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

1.1 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.

1.2 The Learning Path Editor

The Learning Path Editor (LPE)¹ 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.

¹ Available as part of the TenCompetence infrastructure: <http://dspace.ou.nl/handle/1820/2182>

The Learning Path Editor opens in the Master view (Fig. 1.1), 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.

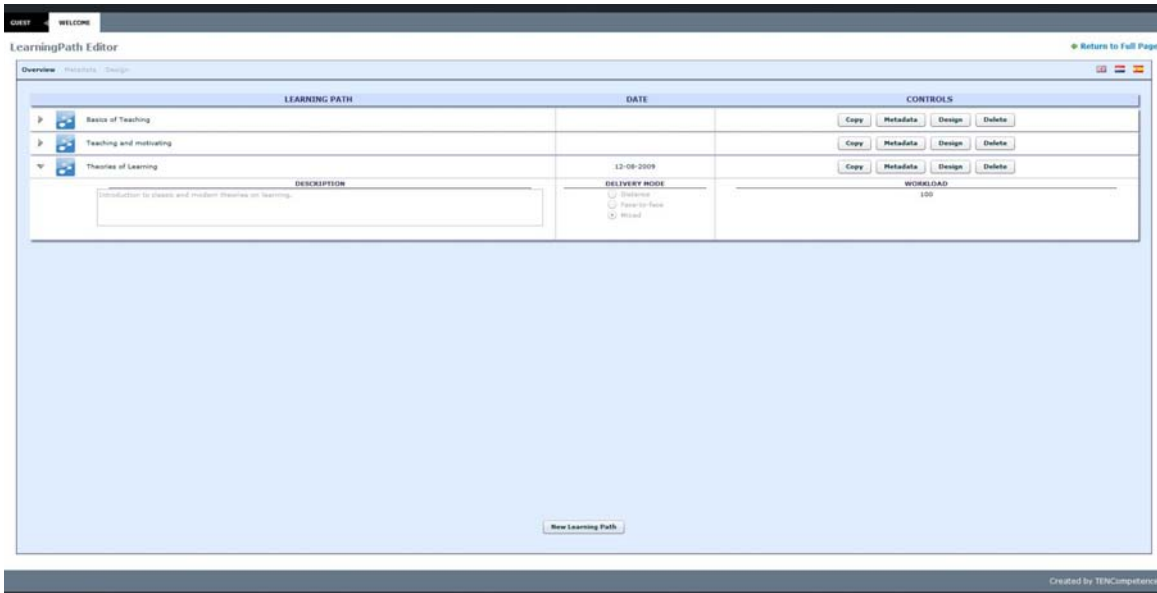


Fig. 1.1. Learning Path Editor Master view

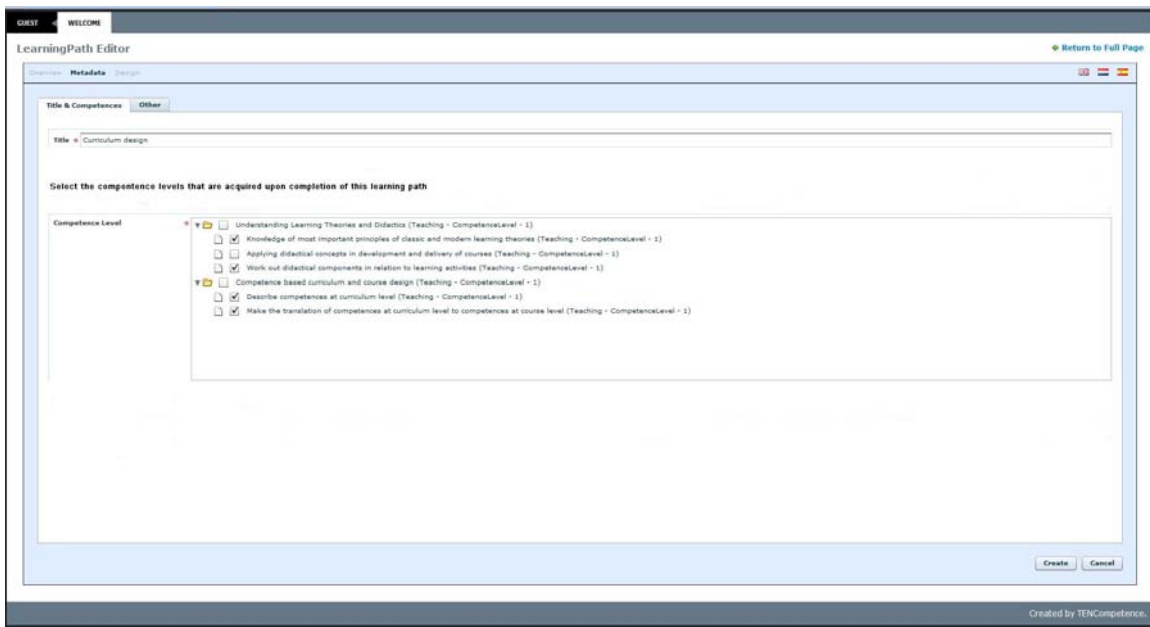
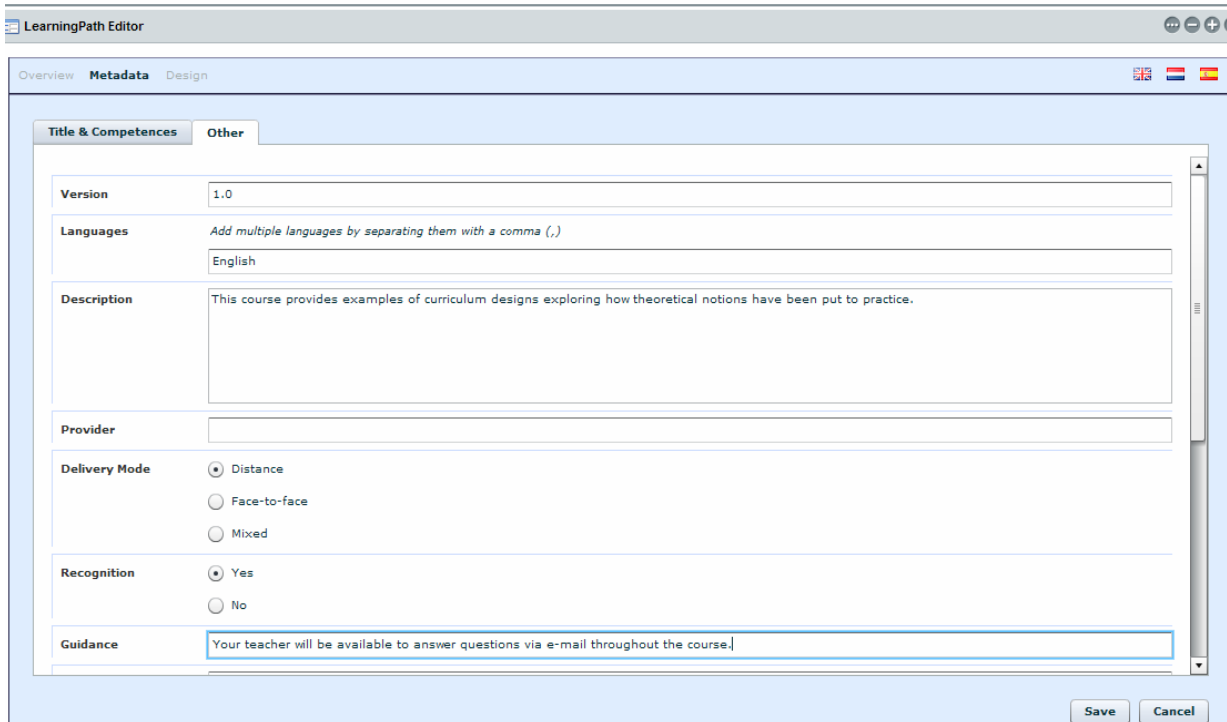


Fig.1.2. Learning Path Editor Metadata view (Title & Competences)

By clicking the button *New Learning Path* the metadata view is activated (Fig. 1.2). 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. 1.3.) 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' window 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. 1.3. 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. 1.4).

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. 1.5 to 1.9). Note that the user has to specify whether the learning path (top level) constitutes a sequence, free order or parallel grouping of learning actions.

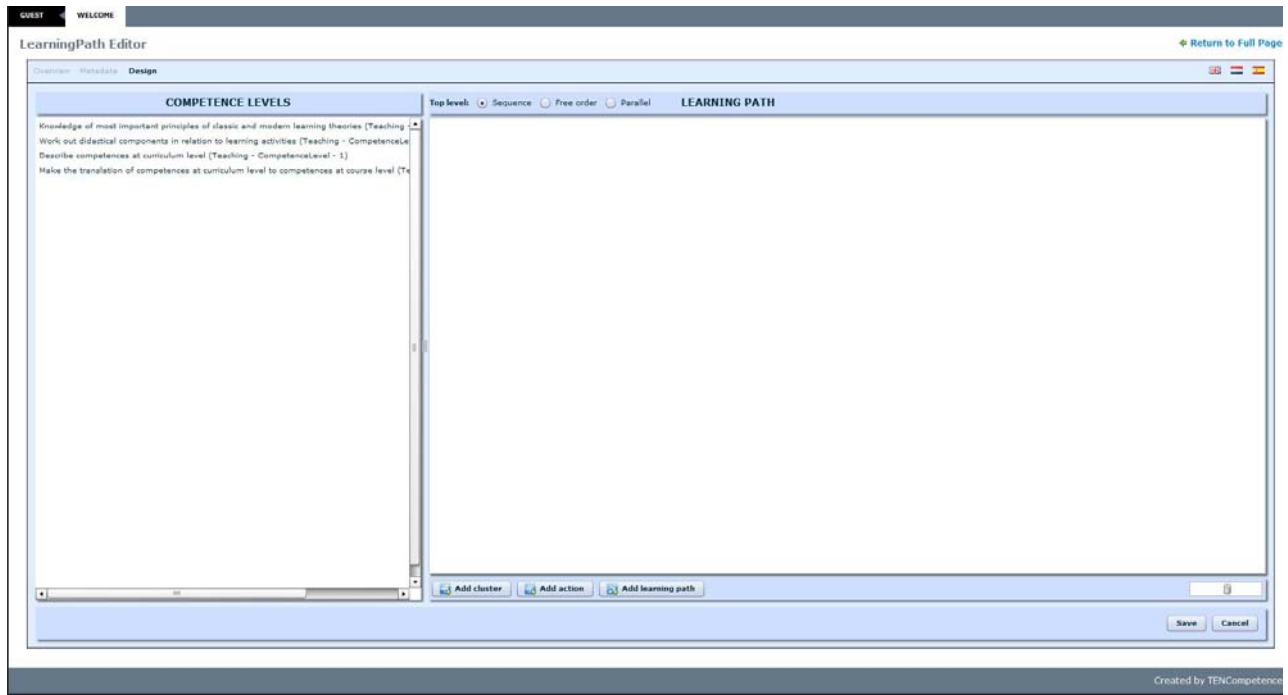


Fig. 1.4. Learning Path Editor Design view

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. 2.5). 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.

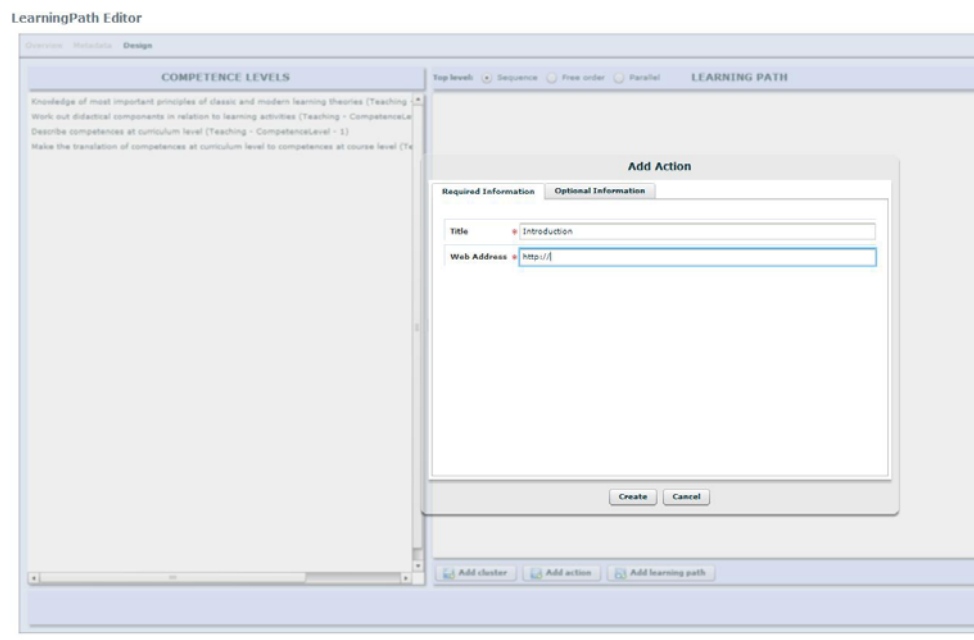


Fig. 1.5. Learning Path Editor Design view: Add Action

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. 2.6). 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.

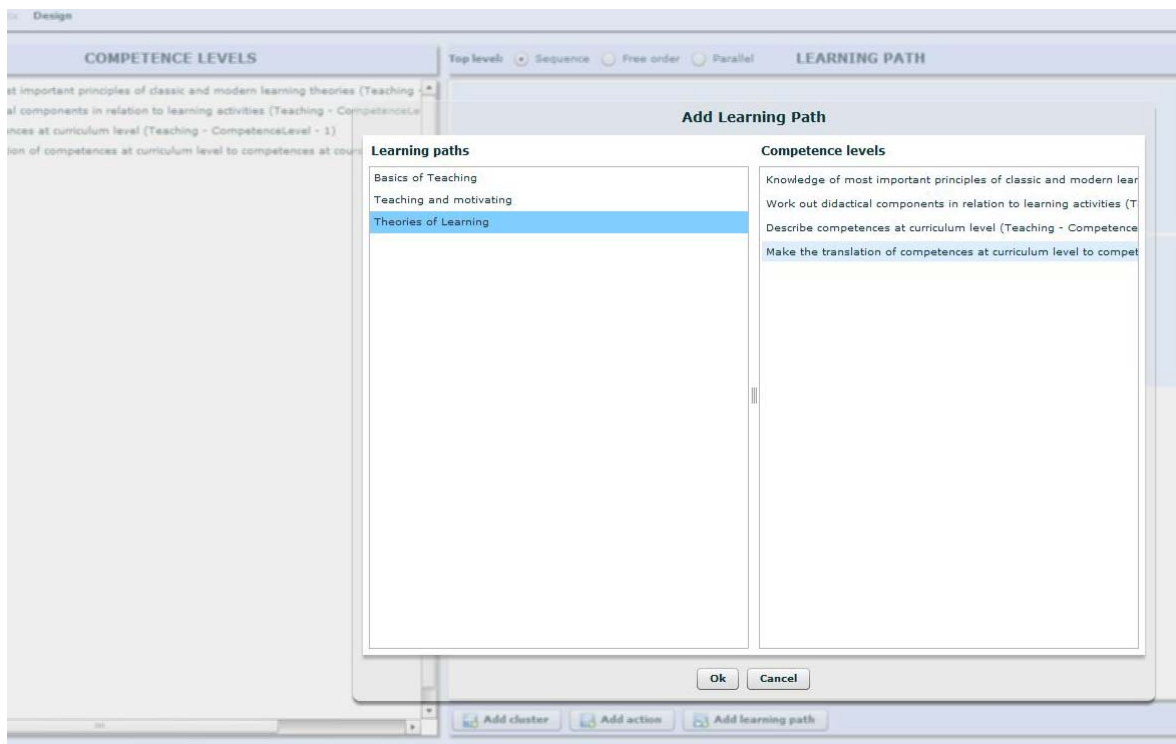


Fig. 1.6. Learning Path Editor Design view: Add Learning Path

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. 1.7.).

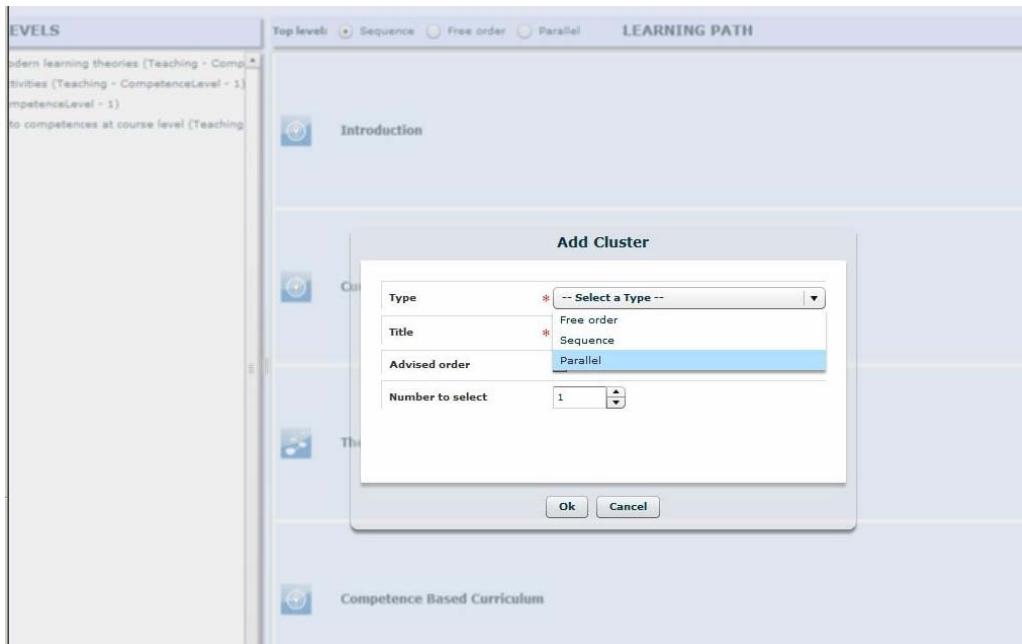


Fig. 1.7. Learning Path Editor Design view: Add Cluster

Once the cluster has been added the author can drag and drop the required actions and paths to the cluster. Fig. 1.8 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 (ID3.24 “Design & architecture of the integrated system”) provides information on the Learning Path Editor in relation to the wider system.

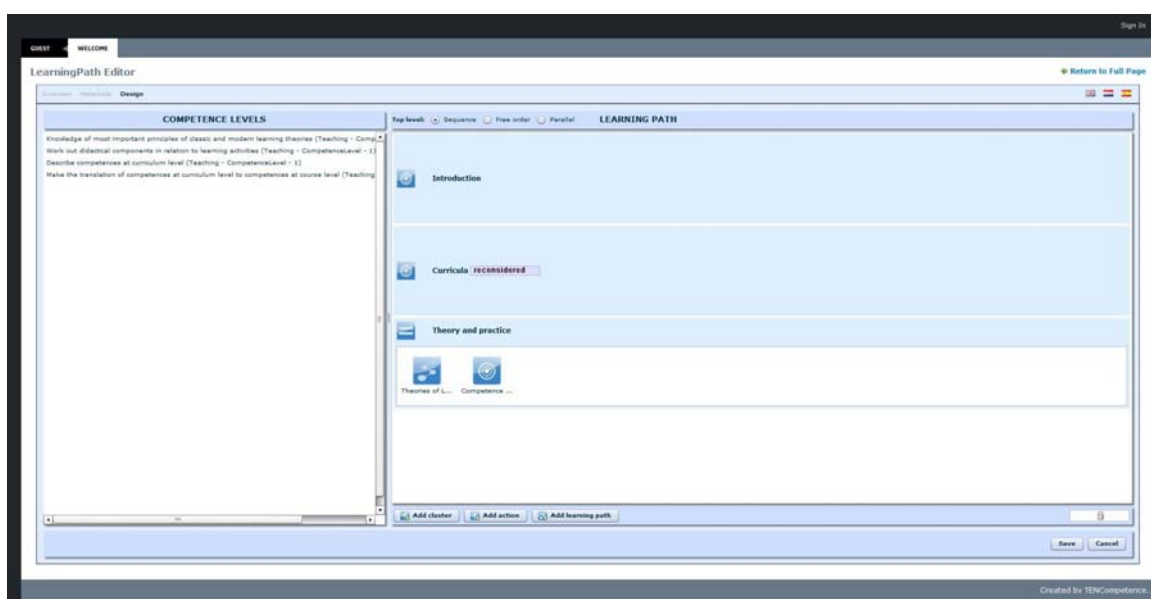


Fig. 1.8. Learning Path Editor Design view: action and learning path included in parallel cluster

1.3 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).

1.4 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.

References

- Beck, B. (2002). Model evaluation and performance. In A. H. El-Shaarawi & W. W. Piegorsch (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>