How to Integrate Learning Design into Existing Practice

DOI:
10.1007/3-540-27360-3_15

Document status and date:
Published: 06/07/2005

Document Version:
Peer reviewed version

15 How to Integrate Learning Design into Existing Practice

José Janssen, Henry Hermans

Educational Technology Expertise Centre,
Open University of the Netherlands, Heerlen, The Netherlands

15.1 Introduction

What does it take for an institution to adopt the Learning Design specification (LD 2003) for the design, development and delivery of its courses? What are the implications at an organizational level? These questions will be addressed in this chapter, drawing on the experiences gained at the Open University of the Netherlands (OUNL) with the deployment of EML. EML is the XML-based Educational Modelling Language developed at the Open University and later integrated in LD as the basis for the modelling of learning designs (EML 2000).

Although there are clear differences between the EML and LD specifications, which will be described in more detail in the next section, there are also many parallels. These parallels are sufficient enough to consider the process of adopting EML on a large scale as a valid frame of reference for deploying LD within an institution or organization.

The OUNL started using EML on a wide scale within its regular course development process in 2002. At present (March 2004) a total of nine courses are delivered to over 2000 users (students and staff) via the Internet, using Edubox. In addition, several other courses have already seen their life-cycle come to an end, including courses developed with external partners and hosted by the OUNL.

In terms of LD some of these courses represent Level B designs, but most of them include the use of notifications (Level C). The type of courses and the tools used in developing and delivering these courses will be described in greater detail below, in Sect. 15.3.

Having thus described the context and the extent of experiences relating to the use of EML within the OUNL, the remainder of this chapter will address several “how to?” questions, regarding the integration of an educational design specification. These “how to?” questions are presented in chronological order, following the stages in the course[1] development and delivery cycle: design, creation and delivery. The stage of analysis preceding the design stage is considered to have led to the decision to develop (part of) the course within LD and will not be dealt with separately.

Attention will be paid to what we perceive as a continuum between different approaches regarding the development of LD courses, with a "tailor-made" approach at one end of the scale and a "bulk" approach at the other. The variety of possible approaches in using LD and the consequences involved in choosing a certain approach will be considered throughout the chapter.

15.2 EML and LD

Is it justifiable to say that the experiences at the OUNL in adopting EML can serve as a model or guide for the integration of LD elsewhere? It is necessary to recognize that there are some differences between LD and EML, the most important being that:

- EML is a single, all-embracing approach to developing learning experiences, making it possible to model, for instance, all types of questions, whereas LD offers a framework which references other specifications in order to model questions, metadata, etc.
- EML contains a content model, allowing content to be modelled "in EML", whereas LD has no content model, leaving it open as to how (in what format) content is modelled, although XHTML is recommended.

In relation to the processes described in this chapter, these differences are not too significant. EML may be conceived of as an implementation of the LD framework, with specific choices regarding the content model and the use of metadata.

In general, the quality of design and creation tools determines to a large extent the efficiency of business processes and strongly influences the acceptance of learning technologies by teachers. As will become clear, tooling has been and still is a problem. However, this can be said to apply to EML as well as XHTML. So, these problems occur when working with EML as well as LD. However, although the availability of appropriate tools is an important issue with regard to the workflow and processes described in this chapter, we have aimed to consider and discuss this work-

[1] Courses are the smallest unit of delivery at the OUNL. However, courses may consist of more than one UOL. In some cases only parts of the course have been modelled in EML.

flow in terms of the underlying purposes and principles, regardless of specific tools. After all, tools can be expected to change and develop rapidly over time.

15.3 The OUNL Case

In the year 2000, after a two-year period of small-scale experiments, the OUNL launched a more ambitious experiment to explore the use of EML in the course development process as part of a strategy to become a Digital University. The so-called “Start projects” aimed to develop six demonstrable Units of Learning (UOLs) in EML within half a year. The ambitious nature of the project is evident from the fact that the staff involved in the development of those courses were trained on the job in working with EML. In addition, the experiences gained during the design and creation process were intended to result in a detailed description of the work processes involved. Prior to these Start projects only few educational specialists had gained any “hands-on” experience with the design and development of educational materials using EML. Now it was time to broaden the scope and see what it would entail to integrate EML in the organization, working with a team approach to course development, as is common with distance education institutions. Educational specialists, subject experts, editors and graphical designers received training and worked together to search for the most efficient ways to get the job done. The training involved both an introduction to EML and gathering hands-on experience with the tools used to create, edit and store EML documents: Framemaker+SGML and Microsoft’s Visual SourceSafe. The tools used in the R&D phase were transferred, without finetuning, to the production environment. At this stage, a stronger division between design and creation appeared. Rather than having the educational specialist (incorporating EML expertise) doing all the “EML work”, subject experts and support staff contributed as well.

Different teams developed different approaches. While some teams focused on the elaboration of a fully explicit pedagogical design before creating the corresponding structures in EML, others chose a more incremental approach. Some paid meticulous attention to the use of metadata, whereas others completely disregarded the issue of metadata. At the time, little integration within existing practices could be identified: the focus was on demonstrable products rather than courses to be delivered.

Meanwhile, several experimental implementations were also set up outside the OUNL. The development of a full curriculum at the Hotel Management School in Maastricht is particularly noteworthy. The separate modules of this curriculum were to be based on the pedagogical concept of competency learning. However, the lack of a common development approach and corresponding templates led to the production of a broad variety of learning designs. Time and money constraints forced the team to switch to a different development approach, in which design flexibility was restricted and a single design template was used to create the remaining courses. As a result of the conventions and rules underlying the (EML) template, the time required to create a concrete module design was reduced significantly.

Another noteworthy implementation is the Law-On-line project of the Digital University (DU), a consortium of universities and institutions for higher education in the Netherlands. The Law-On-line project aims for the joined development of on-line learning materials in a broad variety of law disciplines. These learning materials have a strong focus on self-assessment and are to be used within educational institutions using different delivery systems. Considering the collaborative development and the explicit purpose of reuse, this project has paid extensive attention to the use of metadata.

Back at the OUNL, from 2002 onwards, a step forward was made in the development of nine courses, modelled using EML. The courses were part of the regular curricula of several faculties. To support this major deployment a new version of the runtime system, Edubox, was developed by a software developer, based on specifications provided by the OUNL. In September 2003 the new player was put to use, delivering nine courses, to a total number of over 2000 students and staff members. Five of these courses include the use of notifications (LD Level C), whereas the other courses match LD Level B.

The integration of the EML/LD player into the existing virtual learning environment marks a shift in the functional use of the virtual learning environment. This shift may be characterized as a move from a predominantly supporting function, with a strong focus on information service, towards regulation of the primary educational process for both students and staff.

In conclusion we can say that the OUNL has moved from a pioneering stage towards a stage of consolidation. From all the different approaches and experiences gained hitherto, several recommendations have come to the fore, which will be discussed in the next sections. However, in our view the OUNL still has not reached a stage of full deployment/integration. In particular, authoring tools and processes need to be improved in order to gain more widespread acceptance within the organization and to be able to increase production efficiency. Nonetheless, the experiences gained in pioneering and experimenting offer a considerable empirical base from which guidelines may be derived regarding the adoption of LD within an organization.

15.4 How to Get Started

A major issue relating to the implementation of LD is the extent to which pedagogical flexibility is allowed within the (educational) organization concerned. If the organization is to allow the use of a broad variety of pedagogical concepts and models, this calls for a different approach and tools from an organization which wants to restrict the number of pedagogical models used.

The following scenarios illustrate the possible implications of both approaches, which may be taken to represent two extremes on a continuum.

In the “restricted” scenario the organization is likely to have a highly standardized approach to course development and delivery. Taken to its extreme, there is only one pedagogical model, e.g. problem-based education, and all courses are built in the same way: presenting a set of problems which need to be solved in several steps. Subject experts merely need to have a tool at their disposal whereby they can specify the problems and steps. Relevant materials and services can additionally be selected from a fixed set of resources, ranging from Internet sources to mail services. The tool presents a well-defined learning design, a form for completion, as it were, with a limited set of options to select from and ‘blanks’ to be filled by the subject expert. Subject experts can work relatively “undisturbed”; they need not know that “underneath” (in LD terms) the problem is an activity sequence and that each step represents a learning activity – nor need they consider other possible ways to model a problem-based approach.

At the opposite extreme of the continuum is the scenario resembling the “tailor-made” approach we have witnessed at the OUNL. Whereas in the first scenario a design can be considered to be integrated in the tool the subject expert uses, a tailor-made approach presupposes the flexibility to choose and develop an appropriate learning design. Experiences at the OUNL show that even in this scenario subject experts do not necessarily have to concern themselves with LD specification terminology or with XML authoring tools. Given the circumstances at the OUNL, and the need to work with XML tools in order to keep a wide variety of modelling options open, that side of things was left to the educational specialists, intermediaries trained in EML and the tools used to create it. Working according to this second scenario required considerable finetuning, which will be described in more detail in the following sections. This, of course, cannot necessarily be considered as a viable option in other contexts, where teachers work more individually and independently. A considerable challenge therefore lies in the development of authoring tools which support the design and development of a wide variety of models in a user-friendly way.

In order to get started, appropriate authoring tools, matching the necessary pedagogical flexibility and constraints, need to be chosen or created. Quite conceivably, the whole continuum of approaches needs to be supported. In this case the authoring tool(s) should allow the editing of "basic" LD files, on the one hand, and facilitate the creation of restricted templates, on the other hand (this approach is proposed by the architecture described in Chap. 3).

Once the relevant tools have been selected or created, staff involved will need to receive proper training regarding their use.

15.5 How to Design

During the design phase the outline of the course is planned. In the restricted scenario the design phase will involve matching course parts and content with predefined templates of courses or course-parts. For the tailor-made approach it is recommended that the course design is allowed to evolve in a number of iterative cycles, resulting from close cooperation between educational specialist and subject expert. First, a course outline is created, giving a "full picture" of all course components and the way they relate to one another. This outline is ideally represented schematically. Such a schematic representation ("educational architecture") could be a simple drawing in Word, a UML diagram, or some other, more sophisticated representation from, for instance, the MOT+ graphic editor described in Chap. 9. A schematic is created in order to facilitate communication and discussion between educational specialist and subject expert. It helps to establish whether all elements of the course are "in the picture" (e.g. different types of student and tutor tasks, different types of resources and services used). Although the educational specialist will already be analysing the course in terms of LD concepts, a schematic will describe the course in the terms used by the course itself. After all, it is intended to be a tool for communication between subject expert and educational specialist to establish whether they have a common understanding of the course and its constituent parts.

During the next step the educational specialist (EML/LD expert) translates the pedagogy and components of the course model to LD elements. The specific course model is mapped onto the pedagogical "meta model", which consists of abstract notions such as learning activities and learning objects. Figure 15.1 shows how a course can be considered as a specific example of a pedagogical model, which in turn is an illustration of the pedagogical meta model behind LD.

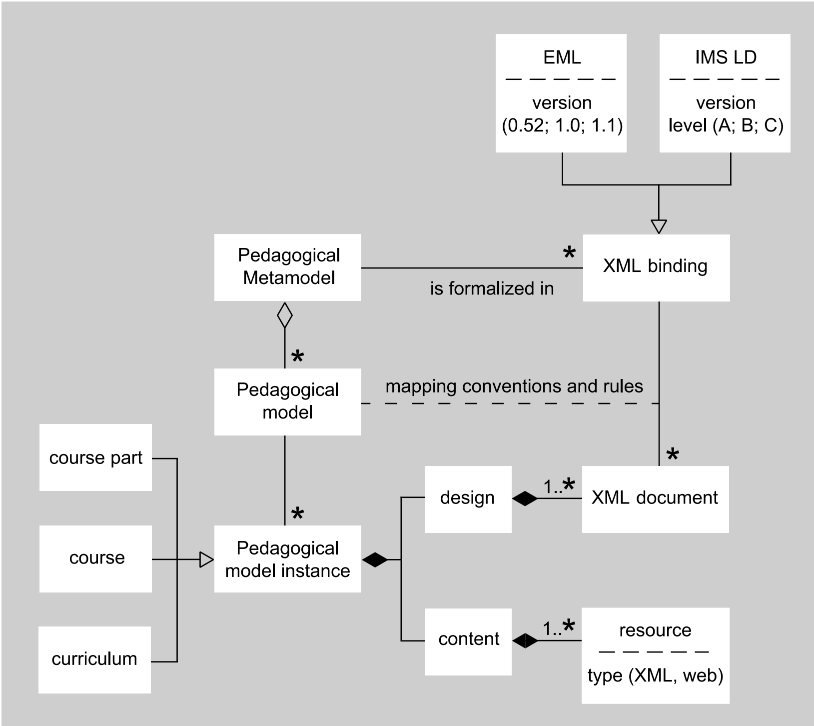

Fig. 15.1. A course as a pedagogical model instance

Figure 15.2 gives a more detailed description of the process of translating the course model to LD elements. First, the course is described in terms of roles, activities (and their inter-relatedness), tools and services. This is done separately for each role. Also, at this stage, decisions are made on the use of metadata, based on considerations regarding reusability: what metadata is needed on which levels? It is necessary to consider these issues at this stage as they may influence the way the course is modelled.

Once the schematic representation is agreed upon and the mapping of the course model to LD is finalized, a prototype can be created, which shows what the course will look like in practice for learners and staff. Thus, a better impression of the different roles within the UOL can be acquired by both the educational specialist and the subject expert.

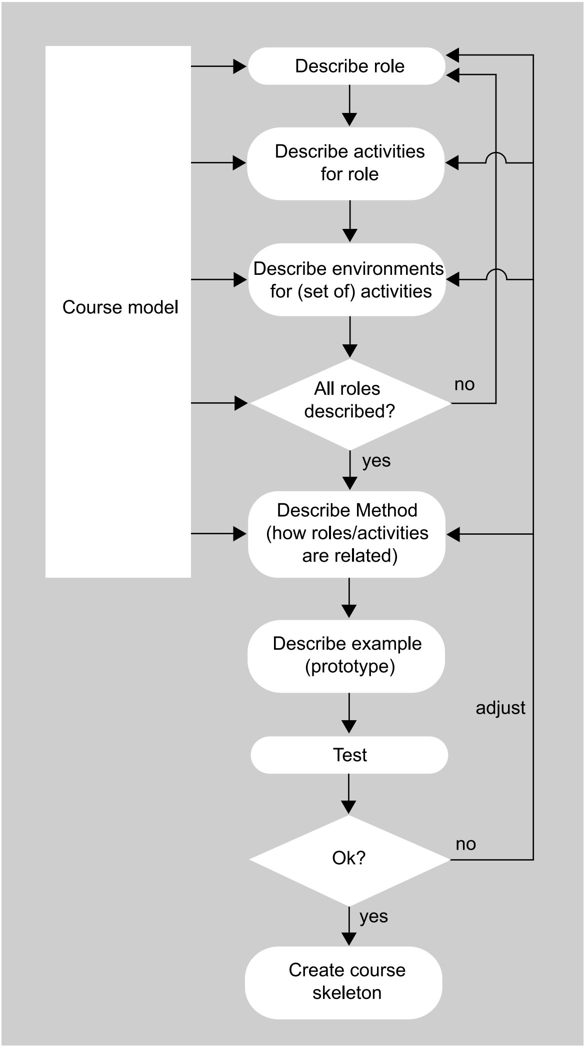

Fig. 15.2. The design process for the educational specialist in detail

So far, the design stage in the tailor-made scenario is a stage in which educational specialist and subject expert come to agree on an increasing level of detail in the design: creating a design is an iterative process, in which previous steps may require adjustments later on, when increased levels of detail may highlight omissions or misunderstandings. Several tools can be used (e.g. schemata, prototypes, etc.) to explain and discuss the design with colleagues who have no knowledge of LD and its concepts.

Once the design is agreed upon, i.e. the model and mapping as specified in the prototype are approved, and each component (e.g. "reading tasks", "exercises", etc.) of the specific course has been identified, it is possible to create the entire course structure in LD, along with templates for components such as learning activities and environments. We recommend that these templates are welldocumented with comments explaining how to use the template and what adjustments are necessary in order to create a new UOL or learning activity. The full skeleton of the course is thus created, which can then be "filled" with content and content-references. This process will be described in the next section. Figure 15.3 summarizes the workflow in the design phase.

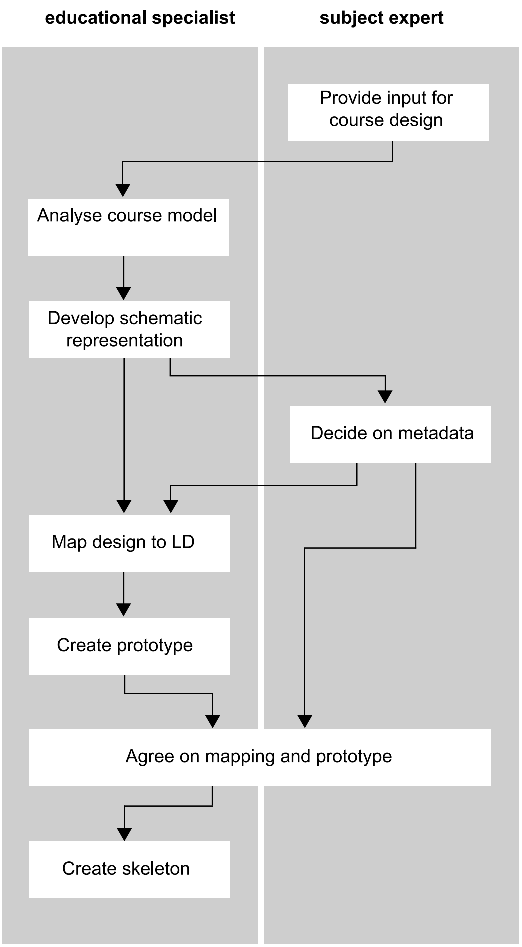

Fig. 15.3. Workflow in the design phase for educational specialist and subject expert

In addition to planning the course structure, arrangements need to be made regarding the storage of content and access to files. Decisions regarding the granularity, storage and management of content will have to be made right from the beginning of the design phase, when the issue of reuse is being addressed. By the time the full skeleton of the course is being created, a contentmanagement system of some kind must be available. However, the availability of a content-management system alone doesn't solve the problem of defining a proper content management strategy. A thorough content management strategy requires that attention is paid to:

1. The domain model, which describes which topics within the broader field of study are covered by what components and how they are related.
2. A metadata model, describing what metadata will be added to which components, and what logic or order will be followed.
3. An authorization model, describing who is responsible for which database files and/or authorized to access which files and to what extent.
4. A "life-cycle" or development model, describing the processes of data entry, publication and testing, correction and updating.

Obviously, this process involves input and agreement from both the educational specialist and subject expert. The subject expert contributes domain knowledge and expertise of classifications and subject indexes used in the field, whereas the educational specialist facilitates the processes of authorization, data entry, publication and testing.

15.6 How to Create

The templates selected (restricted scenario) or created (tailor-made scenario) in the design phase must now be "filled" with content, during the creation phase. Due to the lack of effective authoring tools, three approaches have been adopted at the OUNL:

1. Authors work directly in Framemaker templates which have been prepared for them.
2. Authors work in MS Word and others "copy and paste" to Framemaker. Depending on the complexity involved, these "others" might refer to supportive staff or educational specialists.
3. Authors are given MS Word templates (forms) to work in. This approach is appropriate only for strictly structured content, such as multiple-choice questions. The templates actually consist of EML structures which are hidden with only the relevant input fields (like "question", "correct answer") being visible to the author. This is comparable to the approach used in the Komposer tool described in Chap. 7, although the

approach described there is an alternative way of using word templates. After the form has been completed, the file is converted into an "EML file".

The third approach requires that all formatting (e.g. emphasis, lists, special characters, etc.) is added "manually" afterwards. In the first two approaches authors and data-entry typists receive instructions on the use of these formatting elements, should they have to be used. The approaches in which authors work either "freely" in MS Word or in MS Word templates entail more detailed planning, since extra handling by supportive staff or educational specialists is required. None of these methods of adding content to a design are either effective or satisfactory, illustrating the clear requirement for efficient and user-friendly authoring tools. Working with EML has meant that the OUNL has more or less been obliged to model content in the EML format, whereas LD (or rather Content Packaging) distinguishes between LD content and web content, making it possible, for instance, to simply add Word files. However, this doesn't mean that there are no problems regarding content-authoring in the context of LD. In all instances where content requires learners to produce some input, or where content must be presented in a uniform way (as specified through style sheets, for example) these resources must be created using XHTML, which is currently not supported by adequate, easy-to-use authoring tools.

Once content (including formatting elements) has been added to the design, either directly in EML or via templates, and has been validated, another cycle of evaluation takes place, in order to test the content. This particular stage of testing is comparable to the final editing of written materials and can be carried out by an editor, if they are sufficiently familiar with the content. Errors may result not only from spelling or typing mistakes, but also, for example, from putting a link covering a certain subject in the wrong place. If the editor is not sufficiently familiar with the subject, the testing will have to be undertaken by the author(s). Depending on the complexity of the design and the volume of content modelled this way, the iterative cycle of testing, editing and retesting may take a considerable amount of time, postponing the moment of completion of the course. Once testing has been finalized and any necessary adjustments have been made, the course is ready to be delivered to learners.

It should be highlighted that the process described above presupposes that content creation takes place "beforehand", in design time as opposed to runtime, when learners have already started studying the course. This approach does not necessarily need to be adopted in other contexts, although it is by and large the procedure used within the OUNL. Nevertheless, the delivery system used at the OUNL does allow content to be changed (updated) during delivery (runtime), although it does not allow alterations to the design, such as the addition of entirely new activities. It is important to note that this is simply how the system used by the OUNL is regulated, rather than being an inherent feature of EML or LD.

Figure 15.4 shows the workflow for the roles and tasks involved in the creation phase.

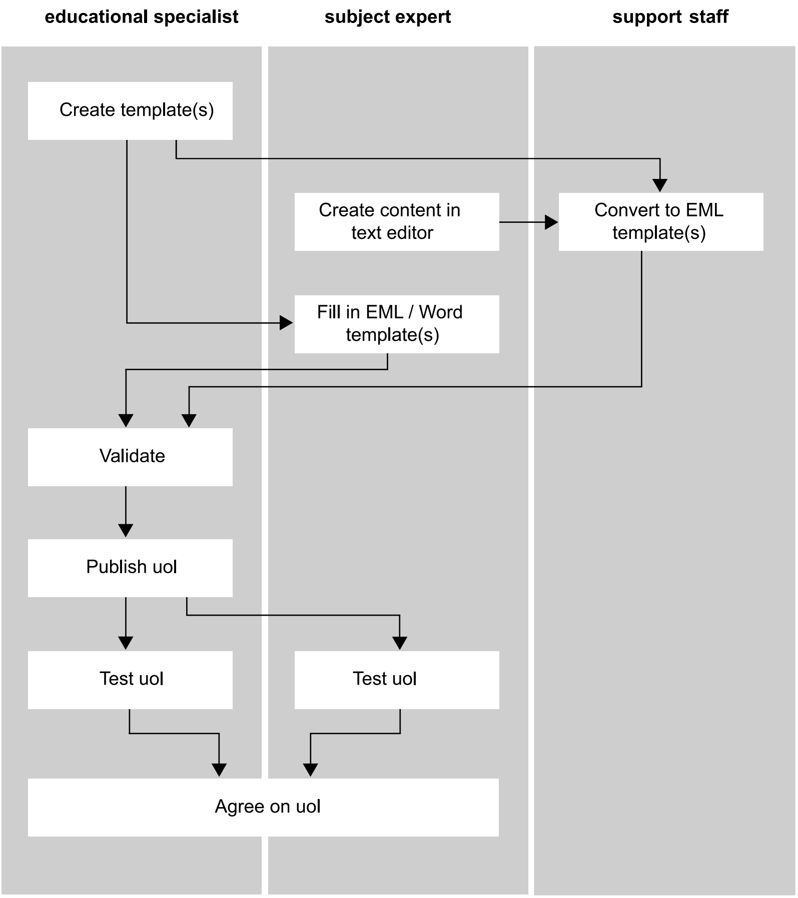

Fig. 15.4. Workflow in the creation phase

15.7 How to Deliver

LD courses at the OUNL are delivered via the web, using a delivery system called Edubox. In implementing a delivery system it is necessary to assess how it will be required to connect with other systems, for instance administrative systems. At the OUNL the delivery system is integrated with an electronic learning environment called Studienet and an interface has been created to connect to the student administration system. Within Studienet students have a personalized homepage; connecting the delivery system with the administrative system has made it possible to automatically add a link to a course delivered by Edubox to the homepages of students who have subscribed to that course. Students gain direct access to Edubox through a "sign-single on" capability, without needing to re-enter their user and password details. Other systems that could be integrated are services such as conferencing clients. However, these connections (interfaces), however relevant, primarily concern practical features of the production environment, rather than the core of the delivery process.

The process of course delivery typically involves a number of actions. The Edubox delivery system consists of two components: Edutool and the player. In order for a course to be made available to students and staff in the player, the course must first be published and users assigned to it. This involves several steps in Edutool:

1. Publication management: first the course must be published. This involves uploading all files and some technical "processing" to check references and materials. Publishing a course also requires a presentation format, which is specified using a style sheet, to be selected. In the player OUNL uses, activities and activity structures are presented in a frame that has the title "To Do". This could be modified depending on the pedagogical style and changed into, for example, "Tasks" or "Problems". It is also possible to provide different style options within a style sheet enabling the interface to be switched to another language.
2. Run management: publishing a course doesn't automatically make it available and visible. Students and staff have to be assigned to the course before it becomes accessible to them. This is done via so-called "runs". This is comparable to a face-to-face course being offered by an institution, where the course has been designed, materials prepared, but the classes have not yet started. Students and staff are 'scheduled' or assigned to the course through a run. Since several runs can be associated with a single version of a course, there is a "create once – use many times" situation. Runs offer a mechanism to spread a large number of students subscribed to the same course over several groups tutored by different members of staff or to organize students into groups according to the study centre they are related to. Run start and end dates can also be set. If a run has no end date specified and the course design or organization doesn't involve a particular grouping, newly enrolled students can simply be added to an existing run. As long as the course (the version) stays the same, it is sufficient to add new students to an existing run or to create new runs. (For more detail on the concept of "runs" see Chap. 4).

3. Role management: after runs have been created and staff and students have been allocated to specific runs, these staff members and students have to be assigned to the specific roles identified in the learning design. It is important to remember that while a course design must include at least one learner role, it may also include several additional learner and staff roles. At this stage of role management the *people* who have been assigned to a run are now assigned to the *role or roles* they will perform while taking or tutoring the course.

At present, at the OUNL, publication management, run management and role management are all coordinated by one person who is in charge of Edutool. However, one could, for instance, also authorize tutors to organize runs and manage roles, although this would require some instruction regarding the use of Edutool.

Other actions that may be necessary to enable the delivery of a course include:

1. Instruction of tutors: depending on the complexity of the design and the variations permitted by the style sheets it may be necessary for tutors to become familiar with the learning design as well as the interface.
2. Services required by the learning design, which the runtime system does not support, may have to be created/instantiated (e.g. communication services).
3. Content update: to the extent that content update may be needed in runtime, arrangements must be made regarding instruction and authorization of those responsible for the updates.
4. A helpdesk service should also be provided, for both students and staff. The need for helpdesk support is likely to vary depending on the scenarios in use. The experiences of the OUNL suggest that with tailor-made scenarios the helpdesk function may become quite complex. Filtering requests for help, in terms of identifying what the problem relates to (the student's computer, provider services, the OUNL learning environment, etc.), becomes more complex with LD, as the delivery system (interface and database operations) and the designs themselves may be potential sources of problems. Consequently, it is recommended that helpdesk staff should get back-up from the educational specialists involved in the design of the LD courses as well as from the staff responsible for Edubox.

15.8 Conclusion and Discussion

Integrating LD within an organization involves a considerable degree of planning, even if we take into account that in future many tasks will be facilitated by increasingly sophisticated and user-friendly tools. The need for additional organization stems from the fact that the deployment of LD introduces new tasks (e.g. related to the publication and authorization of courses), changes current tasks and the tools used to perform them (e.g. design, editing) and may even add to current tasks (e.g. helpdesk support).

In addition to providing staff with sufficient training in order to enable them to adjust to these alterations, the reason for these changes must also be carefully communicated. A notion not uncommon in the field of organizational change states: "As much as possible, necessary skills and favorable attitudes should be fostered *before* changes are introduced" (Johns 1996, p. 565). However, even though permutations on an organizational level may be justified, it may not always make similar sense on the individual level. This is why some level of reluctance or even resistance can be expected in bringing about these changes.

Favourable attitudes require efficient and user-friendly tools. Until these tools are available it is necessary to proceed with care. Even when highly efficient and user-friendly tools have become available, choosing a suitable deployment strategy will be an important first step. The choice between a tailor-made approach, a more restricted approach or a combination of both will also influence the selection of tools. Therefore, in answer to the question "How do you deploy LD within your organization?", our main recommendations are as follows:

1. Decide on the level of pedagogical flexibility/constraint required and choose tools accordingly. Other factors which should be considered include: the degree of (de)centralisation, level of specialization of staff, work processes, the need for runtime flexibility and cost-effectiveness. Generally speaking, allowing more pedagogical flexibility will produce higher expenses, due to the time needed to develop LD courses.
2. Following the guidelines provided in this chapter, consider the workflows involved and decide to what extent they either are supported by the tools chosen, or have been made redundant by increasingly sophisticated tools.
3. Communicate the rationale behind the deployment of LD, the consequences involved for staff and train staff to use the tools chosen.