Alfanet Deliverable D32 – Standards Contribution Report

Project Deliverable Report

Deliverable D32 – Standards Contribution Report

Work package	WP3. Standards			
Task	T33. Standards Contribution Report			
Date of delivery	**Contractual**	30 04 2005	**Actual**	14 04 2005
Code name	ALFANET_D32_ final.doc		**Version** 1.0	draft ☐ final ☑
Type of deliverable	Report			
Security (distribution level)	Public			
Contributors	OUNL, UNED, SAGE, KLETT			
Authors (Partner)	René van Es, Hubert Vogten, Harry Martens, Slavi Stoyanov, John van der Baaren (OUNL); Carmen Barrera, Olga Santos, Jesús G. Boticario (UNED); Maria Jose Carion, Carlos Fuentes (SAGE); Ingeborg Hoke (KLETT),			
Contact Person	René van Es Open University of the Netherlands - OTEC Valkenburgerweg 177 6401 DL. Heerlen		Tel: + 31-45 576.2913 Fax: + 31-45 576.2800 Email: rene.vanes@ou.nl	
WP/Task responsible	OUNL			
EC Project Officer	Mr. Colin Stewart			
Abstract (for dissemination)				
Keywords List	Standards, Learning Design, Question and Test Interoperability, Meta-data			

ALFANET Project Coordination at: Software AG España, S.A. Ronda de la Luna, 4
Tel: +34 91 8079411 - fax: +34 91 8079447 - email: carana@softwareag.es Tres Cantos, E-28760 Madrid

Executive Summary

Introduction

Learning technology standards are widely promoted by various organisations internationally because they potentially have several advantages. Such advantages are for example reusability of learning content, or the reuse of pedagogical scenarios. The current situation is that many learning technology standards have been created and evaluated in limited settings. The aim of the Alfanet project was to use a suite of learning technology standards rather than a single standard to realise adaptive features for learners that otherwise could not have been realised. This has increased the initial investment in development but has allowed us to build an open architecture composed by re-usable components. aLFanet applies important standards in e-learning. The central standard is IMS LD. It enables the design of a variety of pedagogical models and separates the design of the pedagogical model from the content. Thus, it allows to dynamically select from the available learning objects the content to be provided depending on the associated metadata. To complement this standard, IMS Metadata (IEEE LOM) to deal with the knowledge on the contents, IMS LIP for a representation of the user and the IMS QTI to compute the formal progress are used. On top of them, everything is delivered in IMS CP. In addition Alfanet makes use of a set of technical standards in particular SOAP and FIPA.

This report describes the standards used, the Alfanet adaptation scenarios and the role of the standards in these scenarios. The report contains two parts.

In the first part of this report the standards used are introduced and briefly explained and a study on IMS LD, the central standard in the project, is described. Learning Design has been developed to support any kind of education and some prove has been provided by use-cases as presented in the Best practice guide. However, at the time Alfanet selected IMS LD there was no information on how suitable LD is to describe education as it is delivered currently by various educational institutes. Therefore, an investigation was set up to find out what problems would occur if we would try to describe a random sample of lesson plans with LD. Here the findings are reported and special cases that needed more attention are elaborated in more detail.

The second part of the report (chapter 4 and onwards) starts with an overview of the main adaptation scenarios supported in Alfanet. They include adaptation to pre-knowledge, adaptation to learning characteristics, adaptation on assessments and adaptation based on similarities with peers. Subsequently each of these scenarios is explained in a separate chapter and the way the standards are applied to achieve the desired adaptation is clarified. The adaptive features of the Alfanet system are triggered by the data that is collected on individual learners which serves as a base to alter a course for an individual or these data are used to reason upon collective learner behaviour and make alterations in courses for groups of learners. The learner data is stored in a portfolio. This report describes how determining a learners' learning style, cognitive modality and his/her present knowledge on a course specific domain creates the initial learner profile. It is explained how, based on the learners' profile, changes are made to the outline of the course material and what standards are used to realise these adaptations

To determine if a learner makes progress in mastering the learning material, the learner has to take formal assessments during the course, but a learner can also periodically test his/her learning progress. In the Alfanet system tests are used, and two adaptation mechanisms were created that make use of the outcome of these tests. It is explained how question items are created and how adaptive tests can be created out of these individual test items with additional meta-data and specifically created tool. Moreover a description is given how the test results are handled by the Alfanet system to feed them back into the Learning Design of a course.

Besides designed course adaptation the Alfanet system is also equipped with a module that monitors user behaviour and searches in the course material to recommend learner and situation specific interventions that should keep learners motivated and enhance learner performances. It is explained what kind of measures have to be taken at design time to realise this functionality at run-time.

Finally, when courses are designed, the authors have certain expectations with regard to how learners will react to that course. For example, learners are expected to complete a course within a certain amount of time, or authors design a route and learners are expected to follow this route, but are they? To answer questions like these the Alfanet system is equipped with audit functionality that monitors learner behaviour and compares this behaviour with predefined expectations by authors. The results are reported back to the authors with the expectation that when a bottleneck is found in a course it can be put right.

Description of conclusions/results

Alfanet is the first project that integrates five learning technology standards and as such it is unique. During the development of the system and the course materials for the pilot projects several hurdles had to be taken. One of the most challenging one was the integration and cooperation of IMS-LD, IMS LIP, IMS Metadata and IMS QTI, in particular the integration and cooperation of IMS LD with IMS-QTI. Although both specifications come from the same organisation little was known beforehand how such an integration should be realised. But once the exchange of data between the two modules was a fact, it provided multiple possibilities to further refine the adaptive features of the Alfanet system. As a result the current version of the Alfanet system includes a variety of adaptation scenarios based on a tight integration and cooperation of the selected standards. The scenarios can be used directly or as a reference to others involved in systems or standards design and show the feasibility of a generic standards-based framework for e-learning. Additionally, the Alfanet system and/or its components (available as Open Source) are at a stage that they can used for further development and exploration of adaptive-standards based e-learning.

Designing courses using multiple learning technology specifications without an integrated toolset was difficult to manage and time consuming for the authors. The authoring tools for each individual component helped the authors, which were most of the time not experts on the learning technology specifications used, to create their course parts. Nevertheless, it is clear that for real-world usage the authoring process has to be further simplified and more tightly integrated unless there are dedicated experts available to support part of the authoring process.

Table of Contents

1. Introduction

1.1 Situation

At the start of the Alfanet project an analysis was made of existing standards that could be used to realize adaptation. Adaptation in Alfanet is about creating a learner experience that purposely adjusts to various conditions (personal characteristics, pedagogical knowledge, the learner interactions, the outcome of the actual learning processes) over a period of time with the intention to increase pre-defined success criteria (effectiveness of e-learning: score, time, economical costs, user satisfaction). The analysis of existing standards resulted in a set of 6 standards to be used during the development of the Alfanet system.

This report is a reflection on how the recommended standards were used in the project and the contribution each standard delivered to the adaptation in the students' learning environment.

1.2 General overview

This document first summarises the used standards that were identified in D31. Chapters 2 to 6 each describe part of the adaptive functionality of the Alfanet system and elaborate on what standards were applied to reach what functionality.

Chapter 3 describes an investigation in which the functionality of Learning Design was investigated. The aim of this investigation was to find out how well LD is able to describe different types of education as they are presented in lesson plans.

Chapter 5 describes how with use of two test sets the course presentation can be adapted. The first test set described is used to establish the preferred way of learning of a learner and in what format the learning material should be presented. The second test set establishes a learner's pre knowledge for a specific course to determine the course entry point (i.e. the module a new learner can start with). The entry point can be altered such that a learner can skip previously mastered items and can start directly with new items.

Chapter 4 provides a general overview of the functionality that is accomplished by integrating learning technology standards.

Chapter 5 describes what a user profile is and how these were used in Alfanet to personalise course content and course outlines at use time. It is further described how tests specific for this adaptation can be created and used.

Chapter 6 goes into details of tests that are used for various adaptive purposes but that are also adaptive themselves. It first describes how individual test items are created. Then it describes how, by adding more information to the test items, a test can be created of which the presented questions depend on previous learner results. Finally it explains how the results of the tests are exchanged with a course in the Alfanet system to trigger other adaptive features.

Chapter 7 describes how the behaviour of an individual learner and the collective behaviour of all learners in the Alfanet system can be used to support learners in the learning process. It shows that if some additional information is included in a course, software agents can reason about learner behaviour. Also these software agents explore the information that can be found in the services that offered to learners. The combination of this information results in learner advice that enriches or improves the learning process of a learner.

Chapter 8 is about feeding back usage information on a course to its developers. A mechanism is presented to collect the study behaviour of learners and to compare these data with predefined benchmarks. The results are used to generate reports that can be used to locate bottlenecks in a course.

Chapter 9 illustrates what results can be accomplished when all the previously described functionalities are integrated. First a template, that was developed to help course developers, is explained that includes a large part of the adaptive features within a course. Then it is described how this template was implemented to develop a language course and how the adaptive features are realized. This chapter concludes with an illustration of a run of this course, making the adaptive features visible.

2. Used standards in the Alfanet system:

2.1 Overview

In this chapter a summary on the applied standards will be given. It will outline the characteristics of each individual standard. The standards that will be described are:

- IMS-Learning Design
- IMS-Question and Test Interoperability
- IMS-Meta-Data
- IMS-Content Packaging
- IMS-Learner Information Profile
- Other technical standards.

2.2 IMS-Learning Design

The concept of LD is that a learning design can be summarised as follows. A person gets a role in the teaching-learning process, this role can be either that of a learner or staff. For a role outcomes are stated as learning objectives, these outcomes are to be achieved by performing learning activities for learners, or support activities for those in a staff role. If during the performance of activities learning objects or services are needed that these are placed in the environment embedded in the activity. Which role has to perform what activity and at what moment in the teaching-learning process is either by the LD method through conditions or by notifications. The LD model shown in figure 1 is based upon the pedagogical meta model.

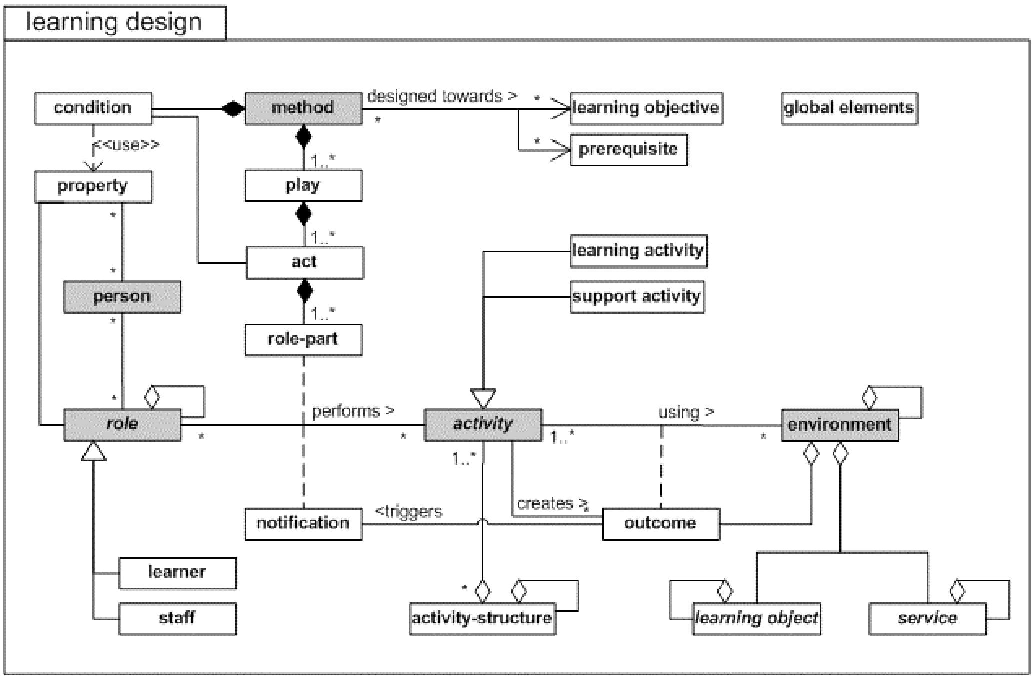


Figure 1. The learning design of a unit of learning.

The core concept of LD, as expressed in figure 2, is that a learning design can be represented by using the following core concepts: A person gets a role in the teaching learning process, typically a learner or a staff role. In this role he or she works towards certain outcomes by performing learning and/or support activities within an environment. The environment consists of the appropriate learning objects and services to be used during the performance of the activities. Which role gets which activities at which moment in the process is determined by the LD method, or by a notification.

In LD the play is placed in the *method* section as shown in figure 2. The LD play contains the acts that have to be carried out in the order as listed. Within an act it is defined who (what role) has to perform what activity or set of activities. As such, the method is the linking pin between all the components of LD; it coordinates the roles, activities and the environments that are associated to the activities. All the other concepts of LD are referenced, directly or indirectly, from the method. The roles within an act link each role to an activity. The activity provides a description of what that role has to perform and what environment is at its disposition. In an act there can be more that one role active at the same time, comparable to the theatre play where there can be more than one actor on the stage at the same time. The activities that are concurrently performed by different roles are synchronized by the act, meaning that if one of the role-parts finishes its activity before the other role-parts, the next act can only become active if the whole act is completed.

```
method
      play*
         act*
            role-parts*
               role-ref
               activity-ref
       conditions
method
```

Figure 2. The method section of LD that contains the play (an asterisk * means that an element may occur more than once).

The method section of LD can refer to these components directly or indirectly:

- Roles
- Activities
- Environments
- Notifications

In LD there are two predefined *roles*, that of a learner and a staff role. Each one of these roles can be further specialized in sub-roles. For example if the course is about designing buildings one learner role could play the role of an architect and another learner could play the role of metal construction expert. Similarly the staff role can be sub-divided. Each role can later be assigned to different activities.

Activities in LD are associated with a role in a role-part, and they contain the actual instruction what a person in that role has to perform. If the activity is directed to a learner and aims to achieve a competence it is referred to as a learning activity. The other possibility is that an activity represents a support activity. Typically support activities are performed by person in a staff role, but learners may also be supported by their peers. Furthermore, activities appear as single activities or they can be grouped in structures such that they have to be carried out sequentially or partially ordered.

Environments is a grouping mechanism for learning objects and services are situated. Learning objects are typically used by learners when performing an activity but these objects make no part of the activity description itself (i.e. dictionaries). Services are used to provide facilities that are helpful in completing activities. Examples of frequently used services are the conference service and the mail service. Environments are linked to activities or activity structures.

There are tree levels (A, B and C) of implementation and compliance in IMS-Learning Design. Level A contains the vocabulary to support pedagogical diversity. All the concepts explained above are part of LD Level A as shown in figure 2. Level B adds Properties and Conditions to level A, which enable adaptation and more elaborate sequencing and interactions based on learner portfolios. It can be used to direct the learning activities as well as record outcomes. Level C adds notifications to Level B. The additions of levels B and C are explained next.

Conditions are placed in the method section and have the form of If-Then-Else rules. The 'If' part of the condition uses Boolean expressions on properties that are defined in the component section. Conditions can be used fine tune the path a learner can take through a course or to adapt a course against some predefined characteristics. For example, a course can be adapted to a learners learning style, showing only visual learning objects to visual learners and verbal learning objects to verbal learners. A course can also be adapted to a learner's prior knowledge, if learner x has prior knowledge on topic y then let this learner start with activity z instead of activity b.

Properties are containers that can store information, for example about a learners' progression in a course (completed activities), a learners' learning style, results of tests, but also learning objects that where added during the teaching-learning process as an outcome of an activity (e.g. reports, papers, video registration of a performance). Properties can be either local or global with respect to the run of a unit of learning. Local properties are only available in a run of unit of learning and they can be used to store data temporarily. Global properties are also available outside a specific run of a unit of learning and are used to store for example data in a learner's portfolio so that it can be used in another run of unit of learning.

```
learning-design
   title
   learning-objectives
   prerequisites
   components
      roles
         learner*
         staff*
      activities
         learning-activity*
            environment-ref*
            activity-description
         support-activity*
            environment-ref*
            activity-description
         activity-structures*
            environment-ref*
      environments
         environment*
            title
            learning objects*
            services*
            environment-ref*
            metadata
method
      play*
         act*
            role-parts*
               role-ref
               activity-ref
metadata
```

Figure 3. The main components of LD Level A.

Besides the condition mechanism, LD Level C also contains a *notifications* mechanism to make new activities available. Notifications can be triggered by a change of a property value, the completion of an activity, a condition that evaluates to true. The notification makes a new learning activity or a new support activity active for a role or it sends a message to person if the person who triggered the notification is not the same as the one that needs to be notified. Notifications can be useful if the input for an activity depends on the outcome of another activity, for example in a collaborative task that is geographically dispersed an the results of a task at location A are used to perform a task at location B.

2.2.1 The unit of learning

The primary use of IMS Learning Design is to model units of learning. A unit of learning is a content package, such as an IMS Content Package, that contains an Learning Design. IMS Content Packages describe their contents in an XML document called the 'package manifest'. The Manifest may include structured 'views' into the resources contained in that package; each 'view' is described as a hierarchy of items called an 'organization'. Each item refers to a Resource that, in turn, can refer to a physical file within the package. It can however also refer to an external resource. Figure 4 depicts the entire IMS Content Packaging conceptual model.

To create a unit of learning, LD is integrated with an IMS Content Package by including the learning design element as another kind of organization within the <organizations> element, using the standard namespace for Learning Design as shown on the right side of figure 4.

The LD element of the unit of learning includes the elements that represent the conceptual model that were briefly outlined before. The details of all the LD elements can be found in the Information model document (IMS, 2003), together with their behavioural specifications.

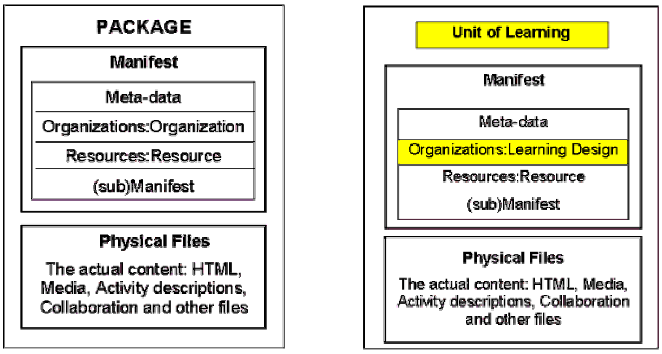


Figure 4.The figure on the left shows the structure of an IMS Content Package. The figure on the right shows the structure of a Unit of Learning, composed by including an IMS Learning Design within the Organizations part of IMS Content Packaging

2.3 IMS-Question and Test interoperability

The IMS Question & Test Interoperability (QTI) specification describes a basic structure for the representation of question (item) and test (assessment) data and their corresponding results reports. Therefore, the specification enables the exchange of this item, assessment and results data between Learning Management Systems, as well as content authors and, content libraries and collections. The specification defines a set of data objects in order to represent assessments, sections and items. These data objects are the base of QTI specification, and they compound the IMS QTI Assessment, Section and Item Information Model (ASI Model).

Despite its name, the IMS QTI specification details more than how to tag questions, tests and results. The standard Question types e.g. multiple choice, fill in the blank, or true/false choice, etc. can be constructed using a core set of presentation and response structures, and results of questions can be collected and scored by using a variety of methods. To represent these options, the IMS QTI specification defines the 'Item'. Items contain all the necessary data elements required to compose, render, score and provide feedback from questions. Therefore, the key difference between a 'Question' and 'Item' is that an 'Item' contains the 'Question', layout rendering information, the associated response processing information, and the corresponding hints, solutions and feedback. Similarly, the 'test' is an instance of an Assessment. Assessments are assembled from Items that are contained within a 'Section' to resemble a traditional test. Additionally, Assessments might be assembled from blocks of Items that are logically related. These groups are also defined as 'Sections' and so Assessments are composed of one or more Sections which themselves are composed of Items, or more Sections. Collectively, these three data objects are referred to as the ASI (Assessment, Section, Item) structures. These evaluation objects can be bundled together to create an object bank. This object bank can then be externally referenced and used as a single evaluation object.

In order to summarize, the main concepts used in QTI specification are:

• **Item** – A combination of interrogatory, rendering, and scoring information;

• **Section** – A collection of zero or more items and/or other Sections;

• **Assessment** – A collection of one or more Sections;

• **Object Bank –** A group of Items and/or Sections that have been bundled e.g. to create an Item-bank;

• **Participant** – The user interacting with an assessment.

As part of IMS QTI Specification the 'Selection & Ordering' was defined. This 'Selection & Ordering' specification contains the description of how the sequence in which Sections and/or Items are presented can be controlled. The selection and ordering process is a two-stage operation in which the child objects are selected according to some defined criteria e.g. meta-data content, etc. and the order of their presentation is then determined.

For adaptive purposes 'Selection & Ordering' defines different rules that can modify the selection criteria, ordering and sequencing in dynamic questionnaires generation:

- Section & Item sequencing. At the current time there are two sequencing rules that can be applied to the selection and ordering rules:
  - Normal. Each object can be presented only once i.e. there is no repetition.
  - Repeat. Each object can be presented any number of times.
- Section & Item Selection. The selection of Sections and Items is based in rules that can be applied to the pool of objects. These rules are applied only to the immediate set of children contained within the parent object. Sections and Items are treated in the same manner i.e. they are objects that may or may not be selected. The following selection rules are supported:
  - All. All of the contained objects are selected
  - Parameterised All. Select all of the objects that have particular properties (the properties are characterized by the object's meta-data.
  - Partial. The random selection of 'x' objects from the set of 'y' objects
  - Parameterised Partial. The random selection of 'x' objects that have particular properties from the set of 'y' objects (the properties are characterized by the object's meta-data)
  - Logical. The logical association of the objects such that the selection of one object is based upon the selection of another. This is supported through the appropriate structuring of Sections and Items.
- Section & Item Ordering. At the current time the ordering of the selected objects is restricted to two mechanisms:
  - Sequential. The presentation in the order in which the objects were selected or occur within the ASI structure
  - Random. All of the selected objects are presented in a random order and the order will change from instantiation to instantiation.

The idea of Dynamic Adaptive Assessments is to generate in run time questionnaires according learner characteristics. Defined rules in course design time will be in charge by selecting what questions will be part of presented questionnaires to the user. In run time rules defined are fed by user characteristics, taking into account that they will be varied over course execution because of learner evolution.

Selection & ordering mechanisms along with user characterization and interoperability with other standards (IMS-LD and IMS-LIP) contribute to create a learner experience that purposely adjusts to various conditions (personal characteristics, knowledge progress) over a period of time, with the intention to increase the success for self-learning..

2.4 IMS-Meta data

A meta-data specification aims to make the process of finding and using a learning resource more efficient by providing a structure of defined elements that describe, or catalogue, the learning resource and requirements about how the elements are to be used and represented.

2.4.1 Components

A meta-data instance is a single specification, that is: a single XML document. This is a 'conforming LOM meta-data instance'. The components of a single meta-data specification are:

- General — Context independent features of the resource. Offer handles for search and retrieval.
- Lifecycle — Features of the lifecycle of the resource. Manage the (change history and) version of the resource
- Meta-metadata — Features of the description rather than the resource. Manage the meta-data entry itself.

- Technical — Technical aspects of the resource. The resource is assumed to be available in an electronic form; a hardback book is hard to describe using this scheme.
- Educational — Educational aspects of the resource. This includes the level of interactivity, for what user (type and level) the resource is intended, and such.
- Rights — Legal aspects of using the resource, i.e. costs and copyright.
- Relation — Possible typed relations with other resources.
- Annotation — Comments on the educational use of the resource.
- Classification — Some classification of the resource, based on a taxonomic path, keywords within the taxonomy.

These components are represented as sub elements of the <lom> root element which is required for all meta-data instances. If the set of constructs is not sufficient, extensions can be made. The proposal suggests the use of alternative namespaces to identify these extensions; it does not provide a strategy.

2.5 IMS-Content Packaging

2.5.1 IMS-Content Packing overview

The IMS Content Packaging portion of the IMS Content framework represents the section that deals with the issues of content resource aggregation, course organization, and meta-data. All of the documents that comprise the IMS Content Packaging specification are focused on the scope represented in Figure 5.

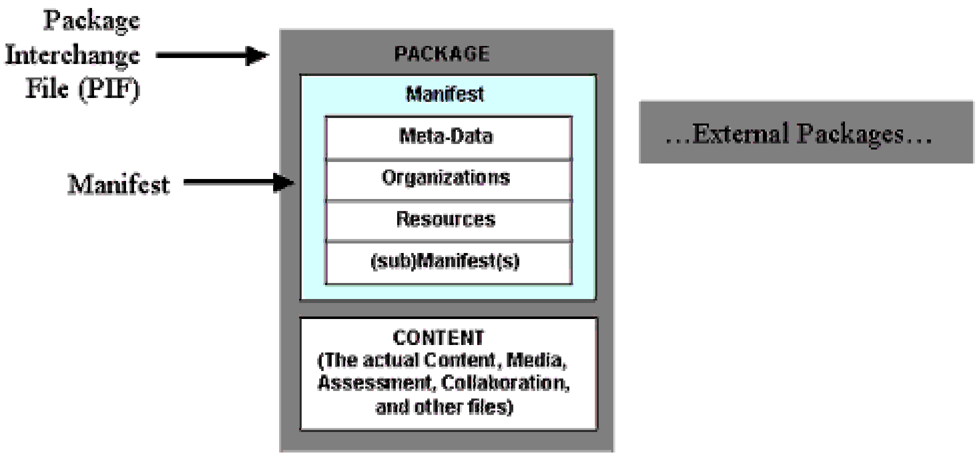


Figure 5. IMS Content Packaging scope

An IMS consists of two major elements: a special XML file describing the content organization and resources in a Package, and the file resources being described by the XML. The special XML file is called the IMS Manifest file, because course content and organization is described in the context of 'manifests'. Once a Package has been incorporated into a single file for transportation, it is called a Package Interchange File. The relationship of these parts to the content container is described below.

2.5.2 Content Package components

Package Interchange File

This is a single file, (e.g., '.zip', '.jar', '.cab') which includes a top-level manifest file named "imsmanifest.xml" and all other files as identified by the Manifest. A Package Interchange File is a concise Web delivery format, a means of transporting related, structured information. PKZip v2.04g (.zip) is recommended as the default

Package Interchange File format Package - a logical directory, which includes a specially named XML file, any XML control documents it directly references (such as a DTD or XSD file), and contains the actual file resources. The file resources may be organized in sub-directories.

Top-level Manifest

This is a mandatory XML element describing the Package itself. It may also contain optional sub-Manifests. Each instance of a manifest contains the following sections:

- Meta-data
- Organizations
- Sub-manifest
- File resources
- Package
- Manifest

These sections will be described below in further detail.

Meta-data section - an XML element describing a manifest as a whole;

Organizations section - an XML element describing zero, one, or multiple organizations of the content within a manifest;

Resources section - an XML element containing references to all of the actual resources and media elements needed for a manifest, including meta-data describing the resources, and references to any external files;

sub-Manifest - one or more optional, logically nested manifests;

File Resources - these are the actual media elements, text files, graphics, and other resources as described by the manifest(s). The file resources may be organized in sub-directories.

Package - A Package represents a unit of usable (and reusable) content. This may be part of a course that has instructional relevance outside of a course organization and can be delivered independently, as an entire course or as a collection of courses. Once a Package arrives at its destination to a run time service, such as an LMS vendor, the Package must allow itself to be aggregated or disaggregated into other Packages. A Package must be able to stand alone; that is, it must contain all the information needed to use the contents for learning when it has been unpacked.

Manifest - A manifest is a description in XML of the resources comprising meaningful instruction. A manifest may also contain zero or more static ways of organizing the instructional resources for presentation.

The scope of manifest is elastic. A manifest can describe part of a course that can stand by itself outside of the context of a course (an instructional object), an entire course, or a collection of courses. The decision is given to content developers to describe their content in the way they want it to be considered for aggregation or disaggregating. The general rule is that a Package always contains a single top-level manifest that may contain one or more sub-Manifests. The top-level manifest always describes the Package. Any nested sub-Manifests describe the content at the level to which the sub-Manifest is scoped, such as a course, instructional object, or other.

2.5.3 Standard Name for the Manifest File

Content distributed according to the IMS Content Packaging specification must contain an IMS Manifest file. To ensure that the IMS Manifest file can always be found within a Package, it has a pre-defined name and location:

imsmanifest.xml

In the absence of this file, the package is not an IMS Package and cannot be processed. It is required that the name be kept, as above, in all lowercase letters.

2.5.4 <manifest> Element

The organization of file resources within a Package is independent of their use. The <manifest> element in an IMS Manifest file serves the purpose of organizing the content for presentation in one or more presentation structures or views and of specifying the resource(s) supporting each view. In this way, a <manifest> element relieves the Package's internal file structure from having to reflect the organization of resources for aggregation or disaggregating. Each resource or set of resources supporting a given presentation view is described for that view, including the path to each file through any internal folders or sub-directories comprising the internal file structure. A Manifest may provide one or more static views of the content.

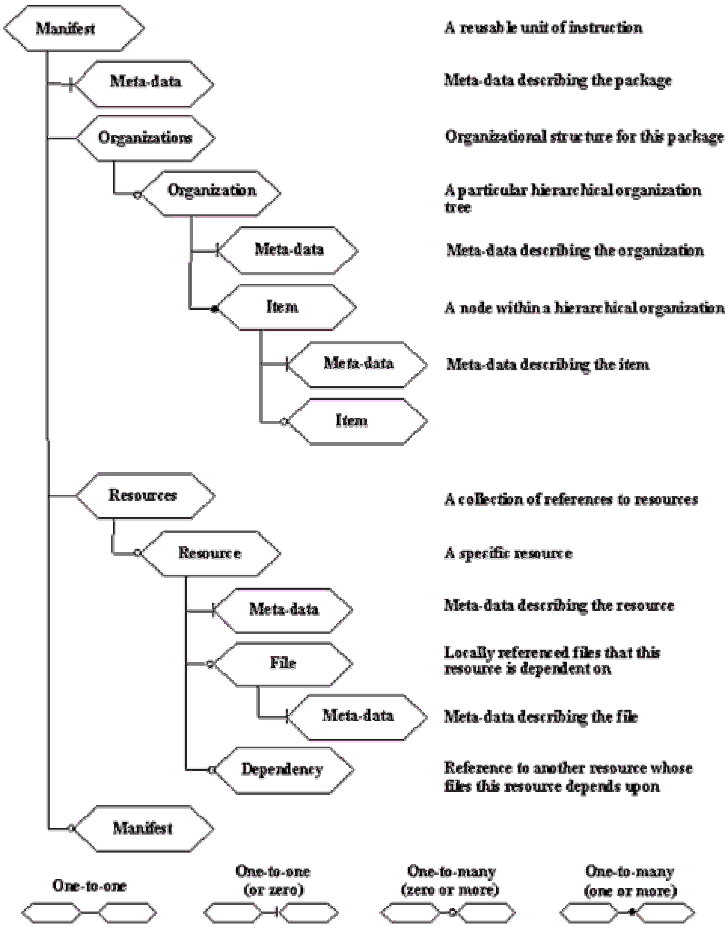


Figure 6. Manifest elements.

2.5.5 <metadata> Element

Meta-data is optional and is allowed in various places in the manifest to more fully describe the contents of a Package. Search engines may look into the meta-data to find appropriate content for a learner or for content repackaging. Copyright and other intellectual property rights are easily declared within the meta-data. Authoring or editing tools could then read the rights stipulated by a content vendor to see if they have permission to open a resource file or files and change the contents.

The IMS CP Information model defines five places where meta-data can be used to describe different components of a content package:

- Manifest
- Organization
- Item

- Resource
- File

If there are requirements to describe any or all of these components with meta-data, then each of these respective components shall be described with separate instances of Meta-data. This construct allows a fine-grained description of each component of a package.

2.5.6 <organizations> Element

There are many ways to organize course or Package content, including no organization at all. In a manifest file, the <organizations> element contains this information.

While many content organization approaches may be developed, a default approach is included as part of this specification. This default approach to content organization, similar to a tree view or hierarchical representation, is encompassed in the <organization> element. The <organization> element is the only element allowed under <organizations>. Content may have additional organization schemas, through the use of the type attribute by setting it to a non-default value. There can be multiple organizations and more than one of the same type, but only one specified as the default.

2.5.6.1 <organization> Element

The <organization> element contains information about one particular, passive organization of the material. The <organization> element assumes a default structure attribute value of hierarchical, such as is common with a tree view or structural representation of data. Future versions of the specification will likely include additional values for the structure attribute to correspond with additional structural organizations or shapes, such as a directed graph, a semantic network, or others. Until additional values are agreed upon, the <organization> element, by default, effectively reads: <organization structure=hierarchical>.

2.5.7 <resources> Element

The <resources> element identifies a collection of content and its files. Individual resources are declared as a <resource> element nested within the <resources> element. A <resource> is not necessarily a single file. It may be a collection of files that support the presentation of the associated presentation structure (<item> element). These files may be internally referenced or externally referenced via a URL. Internally referenced

A <resource> element may also contain a <dependency> sub-element. The <dependency> element identifies a single resource which can act as a container for multiple files that this resource depends upon. Rather than having to list all resources item by item each time they are needed, <dependency> allows authors to define a container of resources and to simply refer to that <dependency> element instead of individual resources. The same restrictions on the values of the identifierref attribute apply to <dependency> as apply to <item> (see Section 4.4.2 for further guidance), with the exception of referring to resources in sub-Manifests. An <item> can do this, a <dependency> can't. Below is an example of using <dependency>.

```
<resources>
  <resource identifier="R_A1" type="webcontent" href="sco06.html">
    <metadata/>
    <file href="sco06.html" />
    <file href="scripts/APIWrapper.js" />
    <file href="scripts/Functions.js" />
    <dependency identifierref="R_A4" />
    <dependency identifierref="R_A5" />
    <dependency identifierref="R_A6" />
  </resource>
  <resource identifier="R_A2" type="webcontent" href="sco1.html">
    <metadata/>
    <file href="sco1.html" />
    <file href="scripts/APIWrapper.js" />
```

```
    <file href="scripts/Functions.js" />
    <dependency identifierref="R_A5" />
  </resource>
  <resource identifier="R_A4" type="webcontent" href="pics/distress_sigs.jpg">
    <metadata/>
    <file href="pics/distress_sigs.jpg" />
  </resource>
  <resource identifier="R_A5" type="webcontent" href="pics/distress_sigs_add.jpg">
    <metadata/>
    <file href="pics/distress_sigs_add.jpg" />
  </resource>
  <resource identifier="R_A6" type="webcontent" href="pics/nav_aids.jpg">
    <metadata/>
    <file href="pics/nav_aids.jpg" />
  </resource>
</resources>
```

2.6 IMS-Learner Information Profile

2.6.1 IMS Learner Information Specifications Overview

The IMS Learner Information Package (LIP) is based on a data model that describes those characteristics of a learner needed for the general purposes of:

- Recording and managing learning-related history, goals, and accomplishments;
- Engaging a learner in a learning experience;
- Discovering learning opportunities for learners.

The specification supports the exchange of learner information among learning management systems, human resource systems, student information systems, enterprise e-learning systems, knowledge management systems, resume repositories, and other systems used in the learning process. In this document such systems will be called learner information systems regardless of any other functionality they possess or roles they fulfil. The IMS Learner Information Package specification does not address requests for learner information or the exchange transaction mechanism.

2.6.2 Information Model

The data model for the LIP is shown in Figure 7

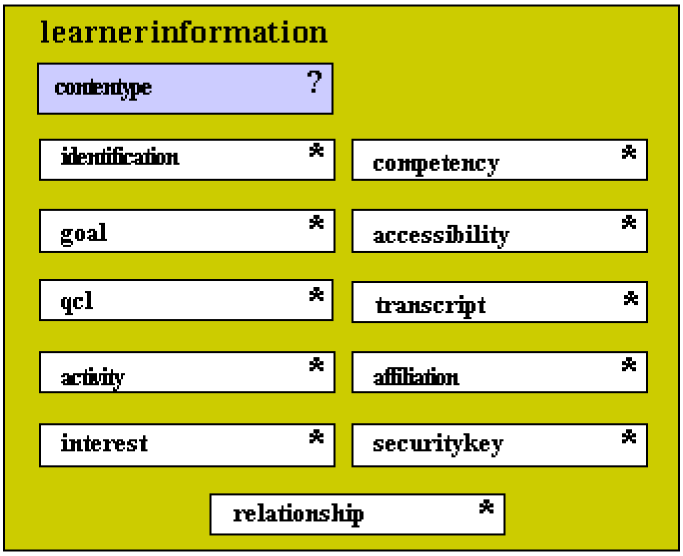


Figure 7. The IMS LIP data model.

The objects in this model and their key behaviours are:

Learnerinformation	The data structure responsible for encapsulating the eleven core learner information classes. The control information describing the learner information as a whole is contained within the 'contentype' class;
identification	The learner information that contains all of the data for a specific individual or organisation. This includes data such as: names, addresses, contact information, demographics and agent;
accessibility	The learner information that consists of the cognitive, technical and physical preferences for the learner, their language capabilities, disability and eligibilities;
goal	This learner information consists of the description of the personal objectives and aspirations. These descriptions may also include information for monitoring the progress in achieving those goals. A goal can be defined in terms of sub-goals;
qcl	This learner information consists of the qualifications, certificationss and licenses awarded to the learner i.e. the formally recognised products of their learning and work history. This includes information on the awarding body and may also include electronic copies of the actual documents;
activity	The learner information that consists of the education/training, work and service (military, community, voluntary, etc.) record and products (excluding formal awards). This information may include the descriptions of the courses undertaken and the records of the corresponding evaluation;
transcript	The summary record of the academic performance of an individual with respect to a particular institution. The transcript is normally supplied by the body responsible for evaluating the performance of the individual;

competency	This learner information consists of the descriptions of the skills the learner has acquired. These skills may be associated with some formal or informal training or work history (described in the 'activity') and formal awards (described in the 'qcl'). The corresponding level of competency may also be defined;
interest	The learner information that consists of descriptions of the hobbies and other recreational activities. These activities may have formal awards (as described in the associated 'qcl'). Electronic versions of the products of these interests may also be contained;
affiliation	This learner information is used to store the descriptions of the affiliations associated with the learner e.g. professional affiliations. A learner's membership of the relevant class, cohorts, groups, etc. undertaken when being educated, trained, etc. should be supported using the IMS Enterprise specification;
securitykey	This learner information is used to store the descriptions of the passwords, certificates, PINs and authentication keys. These keys are used for transactions with the learner;
relationship	The container for the definition of the relations between the other core data structures e.g. 'qcl's and the awarding organisation. This enables the construction of complex relationships between the core data structures;
contentype	The container for the control information that is used to describe the learner information. This information consists of referential, temporal and privacy information and is applied to each of the 'atomic' parts of the learner information structure;
referential	The referential information is used to uniquely identify the learner information record as a whole and the individual data components within that record. These enable each piece of information to be identified. The actual identification system is outside the scope of this specification;
temporal	This information is used to describe any time-based dependencies of the data. This includes information such as the date of creation, time-stamp and expiry date of the learner information. The date/time descriptions are expected to conform to the ISO8601 standard;
privacy	All of the data relevant to the privacy, authenticity and integrity of the learner information is contained within this structure. The actual privacy etc. mechanism and architectures used to support the learner information are outside of the scope of the specification but they interact with the learner information through these structures.

2.6.3 Core XML Schema Tree

The core XML schema tree is shown in Figure 8.

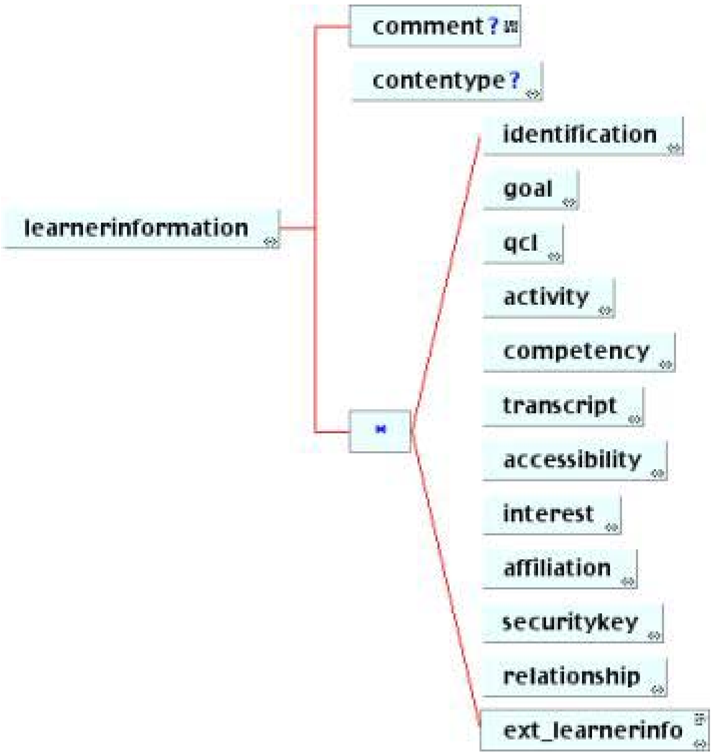


Figure 8. The core XML schema tree.

2.6.4 Activity Product Examples

This example creates an activity containing a product record for the learner. The example contains all of the IMS LIP information required to construct the instance record.

```
<learnerinformation>
   <comment>An example of LIP Activity information.</comment>
   <contentype>
       <referential>
          <sourcedid>
              <source>IMS_LIP_V1p0_Example</source>
              <id>2001</id>
          </sourcedid>
       </referential>
   </contentype>
   <activity>
       <typename>
          <tysource sourcetype="imsdefault"/>
          <tyvalue>Education</tyvalue>
       </typename>
       <contentype>
          <referential>
              <indexid>activity_1</indexid>
          </referential>
       </contentype>
       <date>
          <typename>
              <tysource sourcetype="imsdefault"/>
              <tyvalue>Create</tyvalue>
          </typename>
          <datetime>1980:7</datetime>
       </date>
       <product>
          <typename>
              <tysource sourcetype="imsdefault"/>
              <tyvalue>Coursework</tyvalue>
          </typename>
          <contentype>
              <referential>
                 <indexid>activity_product_01</indexid>
              </referential>
```

```
37              </contentype>
38              <description>
39                  <short>Thesis</short>
40                  <full>
41                     <media mediamode="Text" mimetype="text/word" contentreftype="uri">
42                         myfile/thesis.doc
43                     </media>
44                  </full>
45              </description>
46           </product>
47       </activity>
48    </learnerinformation
```

The key features of this example are:

- The core record for this learner is identified by the <sourcedid> of ‘IMS_LIP_V1p0Example:basic_2001’ (lines 6 and 7). This should be used for other related transactions;
- The type of activity is defined in line 14;
- The product record date of creation is recorded (lines 21-27). The product record itself is given in lines 28-46. The type of product is defined (line 31) and it is given an index identifier (lines 33-37). The product itself, a copy of the learner’s thesis is given in lines 38-45.

2.7 Technical standards

The following standards are used in the AM:

- Simple Object Access Protocol (SOAP: http://www.w3.org/TR/soap/) is used to communicate the AM with the Dispatcher using the : Java API for XML Messaging (JAXM: http://java.sun.com/xml/jaxm/index.jsp)
- FIPA-Agent Communication Language specification (http://www.fipa.org/repository/aclspecs.html) is used for the agents communication. Specifically, the following standards are used:
    - FIPA ACL Message Structure Specification (http://www.fipa.org/specs/fipa00061/SC00061G.pdf) to define the structure of the messages interchanged
    - FIPA Communicative Act Library Specification (http://www.fipa.org/specs/fipa00037/SC00037J.pdf) specifies the semantic definition of the different performative. Accept Proposal, Call for Proposal, Failure, Inform, Propose, Refuse, Reject Proposal and Request are used.
    - FIPA Request Interaction Protocol Specification (http://www.fipa.org/specs/fipa00026/SC00026H.pdf) is used to send the requests to the Coordinator agent, to communicate the Coordinator agent and the Recommendation agents with the Yellow Pages agent and to communicate the Recommendation agents with the Model agents.
    - FIPA Contract Net Interaction Protocol Specification (http://www.fipa.org/specs/fipa00029/SC00029H.pdf) is used to communicate the Coordinator agent with the recommendation agents.

A special effort was done in the implementation of the Interaction Protocols FIPA-Request and FIPA-Contract-Net to facilitate the communication among agents

3. How well can we use IMS-IMS-Learning Design to model educational scenarios

3.1 Introduction

This chapter focuses on the application of learning design in current education. Learning Design has been developed to support any kind of education and some prove has been provided by use-cases as presented in the Best practice guide. However, there is no information on how suitable LD is to describe education as it is delivered currently by various educational institutes. Therefore, an investigation was set up to find out what problems would occur if we would try to describe a random sample of lesson plans with LD. Here findings are reported and special cases that needed more attention are elaborated in more detail. Different pedagogies require different approaches in education and lead to differences in course designs and activities that learners and staff have to perform. Learning designs should offer the flexibility to support all the variances that arise from different pedagogical models. We will also report if the difficulties we found during the coding of the lesson plans are related to specific pedagogies, although these findings can not be generalized due to the sample size.

3.2 Scope of the research

The focus of learning technology specifications was set for a long time on developing specifications for learning objects. A learning object is defined by the IEEE LTSC (2000) as any entity, digital or non digital, that can be used, reused or referenced during technology supported learning. Specifications for learning objects have primarily been designed to ensure interoperability, focusing on technology issues and reuse of learning objects. The instructional value of learning objects is hardly discussed.

Most of the open e-learning specifications that were released for course development and course delivery up to now are limited to a restrictive set of supported pedagogies (Rawlings et all., 2002). If we look at the full spectrum of course development and delivery most specifications focus on the definition of learning objects and metadata and on the sequencing of the learning objects. The result of this narrow focus is that learning is limited to the consumption of content. Teaching is then limited to the art of selecting the right content and putting it in a structured, sequenced way, and of tracking the learner's progress and assessing the acquired knowledge.

To overcome the limits of existing learning technology specifications and standards, the Open University of the Netherlands developed a specification named Educational Modeling Language (EML) (EML, 2000; Hermans, Manderveld, and Vogten, 2004; Koper and Manderveld, 2004). EML provides a pedagogical framework of different types of learning objects, expressing relationships between the typed learning objects and defining a structure for the content and behaviour of the different learning objects. Based on EML the IMS Learning Design (LD) specification was developed and released in 2003. One of the characteristics of LD is its pedagogical expressiveness which can be summarized as the ability to describe units of learning based on different theories and models of learning and instruction together with the learning objects used, adjusted to personal needs.

To date, little is known about the possibility of expressing current educational practice with LD, including both traditional and more innovative forms of teaching-learning situations. This article describes how samples of today's education are taken and it is tested if these samples can be expressed with LD. If there are situations found difficult or impossible to express with LD then these are further investigated to find out if a solution can be found. If there are situations found for which no solution can be provided than these might eventually lead to a change in the LD specification.

3.3 Pedagogical models

During the development of EML a pedagogical meta model was developed. A pedagogical meta model is an abstraction of pedagogical models. This means that pedagogical models could be described (or derived) in terms of the meta model. The reason for developing a meta model was to have a model that was neutral with respect to different approaches of learning and instruction. Neutral in this context means that specific pedagogical models, like problem based learning models or collaborative learning models, should be expressible in terms of the meta model with the same ease.

Models obtained from literature were studied (see Koper, 2001; Koper & van Es, 2004) in three major streams of instructional theories and models (Greeno, Collins and Resnick, 1996). The three major streams of instructional theories are:

- empiricist (behaviourist);
- rationalist (cognitivist and constructivist);
- pragmatist-sociohistoric (situationalist).

These instructional theories have different views on topics such as: knowledge, learning, transfer and motivation. These three streams of instructional theories can be very helpful to map theoretical or practical models of learning and instruction. To evaluate the pedagogical flexibility that was identified above these three major streams were used. To explain how the pedagogical expressiveness can be investigated we need to elaborate the relation that exists between the LD specification and the pedagogical models as shown in figure 8.

The pedagogical meta model is an abstraction of pedagogical models and contains the commonalities found between several pedagogical models. The pedagogical meta model is expressed in a Unit of Learning schema that contains all elements of the pedagogical meta model and restrictions on their usage. The purpose of an XML Schema is to define the legal building blocks of an XML document, just like a DTD. An XML Schema defines elements that can appear in a document, attributes, child elements and their order and their number, defines whether an element is empty or can include text, defines data types for elements and attributes and defines default and fixed values for elements and attributes.

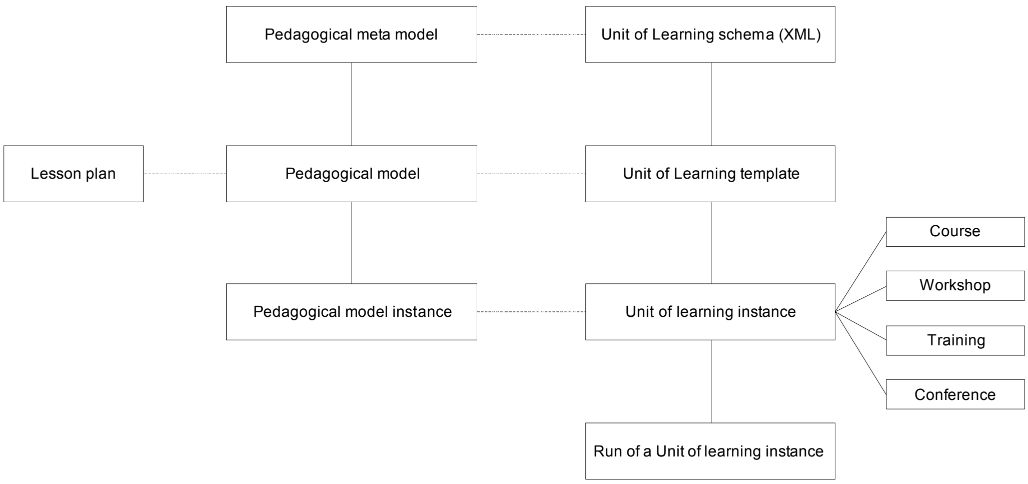


Figure 9. Relation between the pedagogical models and LD.

The LD schema is used to validate instances of units of learning (UOL) that are created with an LD editor. Validation of an instance of a UOL means that the document is checked against the rules stated in the schema, for example that the structure of the document is correct, that multiplicity rules are followed up and the references to learning objects and services are correct. However, the scope of UOL schema is too broad to evaluate because only the correctness of an UOL instance validated, nothing can be said about the meaningfulness of the document for the teaching-learning process. Therefore we take a closer look at pedagogical models that served as input for the development of the meta model.

Pedagogical models were analysed and abstracted to derive the pedagogical meta model. A pedagogical model is defined as a method that prescribes how a class of learners can achieve a class of learning objectives in a certain context and knowledge domain. Pedagogical models are inspired by theories on learning and instruction. Examples of methods are learning Spanish as a second language, how to learn mathematical skills for engineering, or how to plead in someone's defence during a trial. A pedagogical model can be represented as a Unit of Learning template in XML. Such a template imposes further restrictions upon the Unit of Learning resulting in a structure that is unique for a pedagogical model. The rules of a template may for example state that a learning activity is always followed by a self-test and a learning activity always has a conference service defined in the environment. By defining a template, course designers are helped in implementing a specific type of instruction such as problem-based learning.

Closely related to pedagogical models are lesson plans that also describe how learners can achieve a set of learning objectives but in a less restrictive form than pedagogical models do. Lesson plans do not necessarily have a strong relation with learning theories. Teachers who are familiar with a certain topic often create lesson plans for their fellow teachers and make them publicly available.

A pedagogical model instance is an application of a pedagogical model for specific learning objectives and a specific domain. It is more detailed than a pedagogical model in the sense that content and assignments are made concrete. For a Unit of Learning this means that resources are added to the design.

A run of a UOL instance is when concrete learners and staff are assigned to a course and when the time and location are scheduled. If services are defined in the UOL then the applications that handle these services are also prepared with the defined settings. If in the UOL properties are defined then instances of these properties are created in the systems database.

Referring again to the theatre metaphor back into mind, we can compare the pedagogical model to the complete script that outlines the whole play. An instance of a pedagogical model would then be, in addition to the play script, all the stage attributes, the decor, and the lighting. When a run of a UOL is created it means that the play is programmed for a specific theatre, actors are trained to perform the play, tickets are sold to the audience, and the theatre stage is prepared.

For the investigation learning material from current education was used. Current education covers all types of education ranging from primary school to higher education and continuing education. To be able to generalise the results no restrictions were imposed on the type of education. The learning material investigated had to provide enough information, which means that all the aspects found in requirements must also be included in the learning material. For this reason we decided upon using lesson plans as learning material for the following reasons. Lesson plans usually describe how a series of lessons or a single lesson should take place. It is expected that curriculum structures are not more complex that those structures used within a lesson. Lesson plans provide guidelines to developers of learning material based on instructional theories, which have a closer relation to pedagogical models than concrete lesson material. Personalisation is expected to have more impact on materials used within a lesson than it has for example on a course or a curriculum.

3.4 Method

3.4.1 Selection of learning material

We used lesson plans that were available online at 12 web sites (see table 1) in the English language. The lesson plans offered on these web sites covered the full range of education, from kindergarten till university. A total of 16 lesson plans were drawn at random from the selected web sites covering various subjects. We choose for random selection to get a representative sample of lesson plans that are currently used in education. Table 2 shows the lesson plan title, subject and a reference to the web site from which it was drawn.

All 12 web sites that provided lesson plans used subject categories (i.e. mathematics, physics, biology) to present their lesson plans. We followed the procedure as shown in figure 10 to select a lesson plan from one of the web sites.

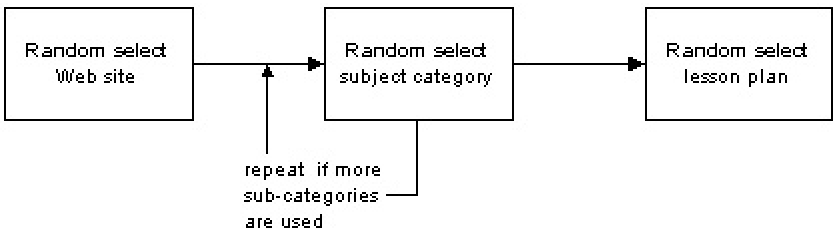


Figure 10. Procedure followed to select a lesson plan from one of the 12 web sites.

For example, first a random number between 1 and 12 was generated to determine the web site where to pick the lesson plan from. Les assume the generated number was 1, then according to table 1 the lesson plan would be taken from web site of The Gateway to Educational Materials. That web site used 12 subject categories (see figure 11) to organise their lesson plans.

Web site number	Web site name	Available lesson plans	URL
1	The Gateway to Educational Materials	36,000	www.thegateway.org
2	LessonPlanz.com	300	www.lessonplanz.com
3	PBS teachersource	4500	www.pbs.org
4	Lessonplan search	2300	www.lessonplansearch.com
5	Merlot	9500	www.merlot.org
6	Statistics Canada	400	www.statcan.ca
7	National Grid for learning	190	www.ngfl.gov.uk
8	Teachers.net	1000	teachers.net/lessons/
9	SMETE	300	www.smete.org/smete
10	Knowledge Agora	350	www.knowledgeagora.com
11	Retanet	65	ladb.unm.edu/retanet
12	National learning network materials	70	www.nln.ac.uk/materials

Table 1: Used web sites that offered lesson plans with an approximate number of lesson plans offered and web address.

Lesson plan title	Subject	Reference
Tongue Twisters	Language arts	2
Lincoln's Secret Weapon	Science & Technology	1
Rhythmic Innovations	Mathematics	3
Consider Copying	Science & Technology	1
The Darien Adventure	History	7
Carnival Safety Success	Language arts	5
Exploring Disability	Drama	2
Ecosystems And Well-Being	Health, Science, Geography	6
Kermit The Hermit	Language arts	1
Inventions	Language arts, Humanities	10
Cracking Dams	Science & Technology	2
The Works Progress Administration And The New Deal	Social studies	3
Learning Microsoft Excel	Science & technology	5
How Do People Express Their Faith Through The Arts?	Social studies	4
Eyes In The Sky	Science & technology	9
A Pittsburgh Memory	Language arts & social studies	13

Table 2. Selected lesson plan with subject the lesson plan covers and reference to table 1 to indicate the web site where the lesson plan can be found.

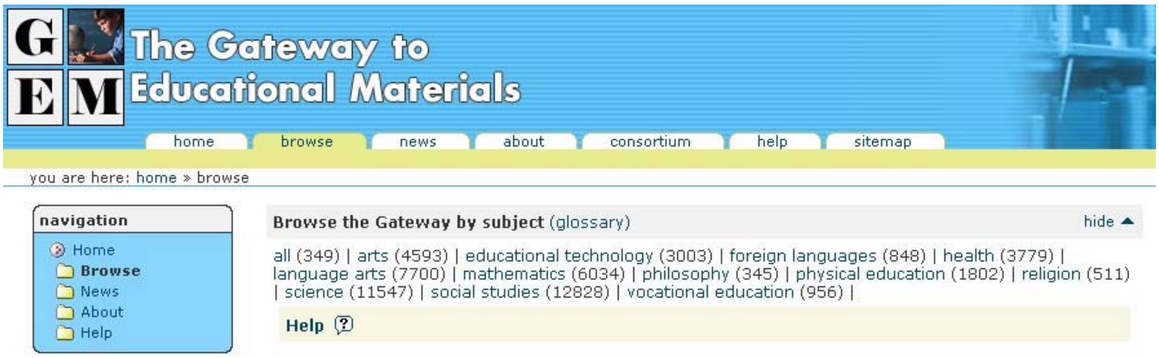


Figure 11. Example of the GEM website where the lesson plans are sorted in subject categories.

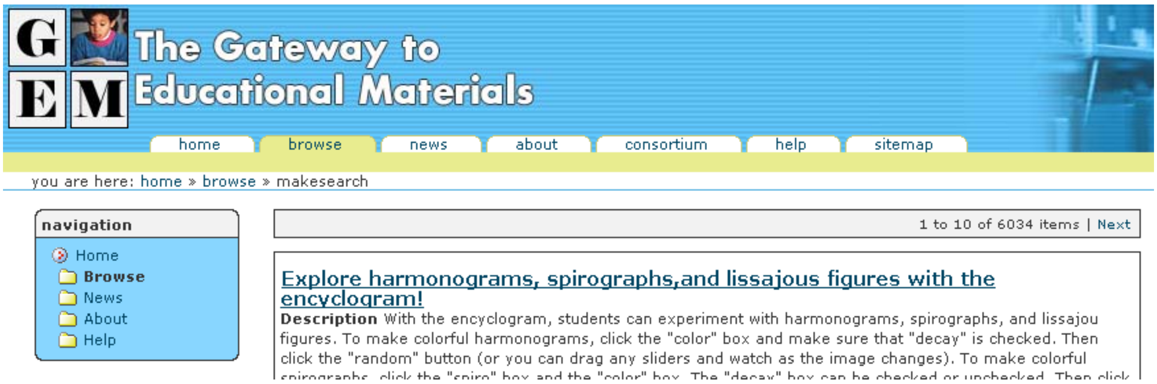


Figure 12. The list of lesson plans that could be found in the mathematics subject category. This category contains a total of 6034 lesson plans as indicated in the red circle.

Next, a random number between 1 and 12 was generated to determine a subject category, for example 6. The sixth subject category from the list is Mathematics and contains 6034 lesson plans as shown in figure 12. Finally a random number between 1 and 6034 was generated to determine the lesson plan that would be analysed.

A lesson plan should meet the criteria of having a study duration of at least 1 hour, and containing 2 or more activities. If a lesson plan did not meet this criteria it was replaced by another one.

3.4.2 Methods used to analyse the lesson plans

To investigate if the selected lesson plans can indeed be expressed fully with IMS-Learning Design, we need to elaborate first on what mean by this. A typical lesson plan describes how a learning objective or a set of learning objectives can be reached by learners. A lesson plan is written for a teacher or an educational developer and describes what activities learners and teachers have to carry out, the order in which the activities have to be carried out, the circumstances under which the activities have to be carried out, how learners have to be grouped, and what materials or technology may be used.

We used several criteria to determine to what extent the lesson plans could be expressed in IMS-Learning Design. First, it should be possible to make a match between the concepts found in the lesson plans and the conceptual vocabulary of Learning Design (LD information model, 2003). With this criteria the static structure of the lesson plan is mapped onto LD and if learners or teachers are working on activities in parallel the workflow is synchronised. Second, the workflow laid down in the lesson plan have to be realized with either the constructs of the conceptual vocabulary (i.e. acts and role parts) or by using conditions and properties. The use of acts only provides means to realize a linear workflow, if a more dynamic flow is needed, conditions and properties can change the visibility of most of the elements of the conceptual vocabulary with the exception of an act. Also if some kind of adaptation or personalisation was identified in the lesson plan together with elements of the conceptual vocabulary, the addition of properties and conditions should suffice to realise the lesson plan. Finally, when learners or teachers need to be informed when a certain event took place or a trigger has fired that either a learner or a teacher has to undertake action, than this LD has to provide this.

Several methods were used to analyse the lesson plans, since this was the first time such an investigation was held, we also had to find out what method would provide enough information with the least effort.

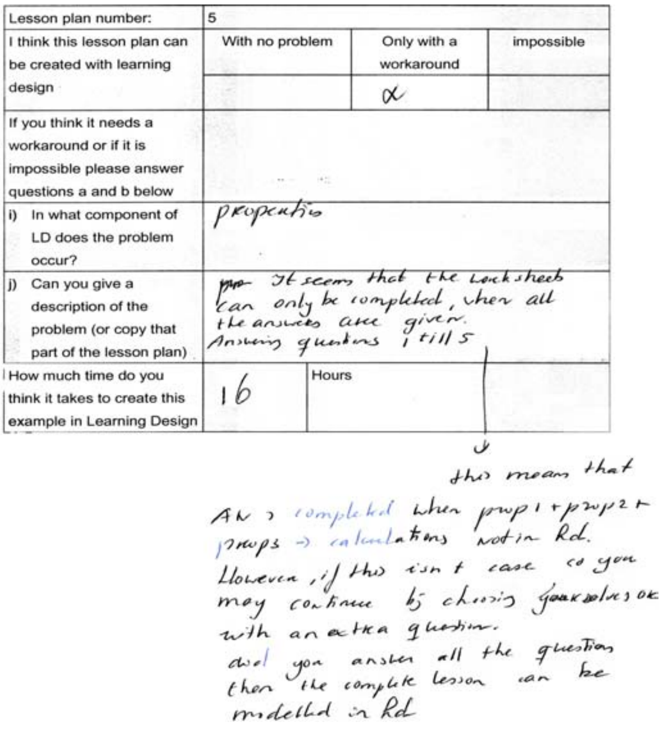
These methods were used to analyse the lesson plans:

- o Expert analysis
- o Document validation
- o Learning Design coding

These methods reported if any situation was found that did not meet one of the criteria. Such a situation could be labelled as a recoverable error or as a non recoverable error. A recoverable error means that something was found in a lesson plan that could not be matched with the conceptual vocabulary, a condition or property was needed for which there was not a clear handle, or a notification had to be given for which there was no trigger provided. A recoverable error is seen as a weakness in learning design that might call for a change or addition to the model. A non recoverable error means that it was not possible to describe a part of a lesson plan with learning design at all, this is considered a shortcoming in learning design.

3.4.3 Expert analysis

This analysis method makes use of experts that asked to give their judgement on how easy or difficult it is to create a learning design instance of a lesson plan at hand. These experts should have extensive experience in learning design coding and must be aware of the possibilities the specification offers. We used two LD experts from the Open University of the Netherlands. The experts were asked to rate a lesson plan on a three-point scale ranging from no problem, recoverable error to non recoverable error. The experts received brief instructions how to carry out the rating but they did not receive any training prior to their rating.



Lesson plan number:	5		
I think this lesson plan can be created with learning design	With no problem	Only with a workaround	impossible
		x	
If you think it needs a workaround or if it is impossible please answer questions a and b below			
i) In what component of LD does the problem occur?	properties		
j) Can you give a description of the problem (or copy that part of the lesson plan)	It seems that the worksheet can only be completed, when all the answers are given. Answering questions 1 till 5		
How much time do you think it takes to create this example in Learning Design	16	Hours	

Figure 13. Example of a lesson plan analysis carried out by an LD expert.

When they identified a recoverable error, or if they encountered a non recoverable error, they were asked to indicate the part of the lesson plan that lead them to their judgement. Figure 13 provides an example of an expert analysis.

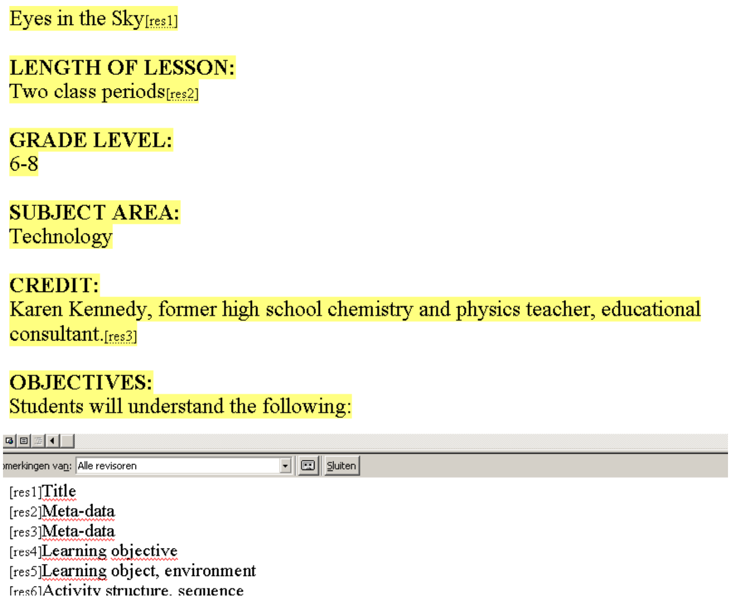
3.4.4 Document analysis

This method uses a set of procedures to make valid inferences from text. Traditionally this method has been used in the social sciences to compare texts and search for relationships between them. Here we don't want to compare text documents, we use it this method to find similarities between the text in the lesson plan and the LD specification. A central idea in content analysis is that words of the text are classified into much fewer content categories. (Weber, 1985). Each category may consist of one, several, or many words. Words, phrases, or other units of text classified in the same category are presumed to have similar meanings. The purpose of content-analysis here is to classify parts of a lesson plan according to the vocabulary used in IMS

Learning Design. This results in a list of categorized text and a residue. This residue is further analysed for elements that do not fit within the pedagogical meta-model but are relevant to the educational model. Residues are thought to be good indicators for a lack of fit of learning design.

The procedure followed involved three iterations that had to be carried out by manually. First the whole text was read and when encountered text blocks that contained words that could be classified that block would be marked. Secondly, the marked blocks of text were further analysed to further classify the text to the concepts of the LD vocabulary. Once the whole text was analysed, the unmarked text became the topic of analysis because for that part no element is available in IMS Learning Design. Further analysis had to reveal if a workaround could be found. A fraction of a lesson plan that was analysed using this method is shown in figure 14.

The lesson plans that were analysed were also classified according to main streams of instructional theories (i.e. empiricist, rationalist, pragmatist-sociohistoric) as described above.


Eyes in the Sky[res1]

LENGTH OF LESSON:
Two class periods[res2]

GRADE LEVEL:
6-8

SUBJECT AREA:
Technology

CREDIT:
Karen Kennedy, former high school chemistry and physics teacher, educational consultant.[res3]

OBJECTIVES:
Students will understand the following:

merkingen van: Alle revisoren Sluiten

[res1]Title
[res2]Meta-data
[res3]Meta-data
[res4]Learning objective
[res5]Learning object, environment
[res6]Activity structure, sequence

Figure 14. Fragment of an analysed lesson plan where in the upper part the original text is shown with the text marks referring to the concepts of the LD vocabulary that are shown in the lower part of the figure.

This data will be used to investigate to what extend difficulties in expressing lesson plans with LD are specific for particular pedagogies.

3.4.5 Learning design coding

The third validation method involves that the lesson plans are transformed a UOL. To do this we followed the procedure described in the best practice and implementation guide of LD (IMS , 2003). The phases in this procedure are:

1. In the analysis phase, a concrete educational problem (use case) is analysed. The analysis results in a didactical scenario that is captured in a narrative, often on the basis of a checklist.
2. The narrative then is cast in the form of a UML activity diagram in order to add more rigor to the analysis. This is the first design step. The UML activity diagram then forms the basis for an XML document instance that conforms to the LD spec. This is the second design step.
3. This document instance subsequently forms the basis for the development of the actual content (resources) in the development phase. The content package with both the resources and the learning design will then be evaluated.

The first phase in the design process is considered completed by selection procedure of the lesson plan. Lesson plans provide detailed description of what a lesson should look like. The next phase in the process is the creation of an activity diagram based on the lesson plan. Figure 11 an example of an activity diagram. What is shown in the diagram are the activities, organized per actor in a so called swim lane. In a swim lane all the activities that a role has to carry out are listed sequentially. The flow through the whole diagram is indicated by start node at the beginning and an end node indicating when the lesson is completed and lines that connect the activities. Activities that are placed at the same horizontal level are carried out at the same time but by different roles. An example of such an activity diagram is shown in figure 15 (Appendix 1 contains the activity diagrams of all 16 lesson plans).

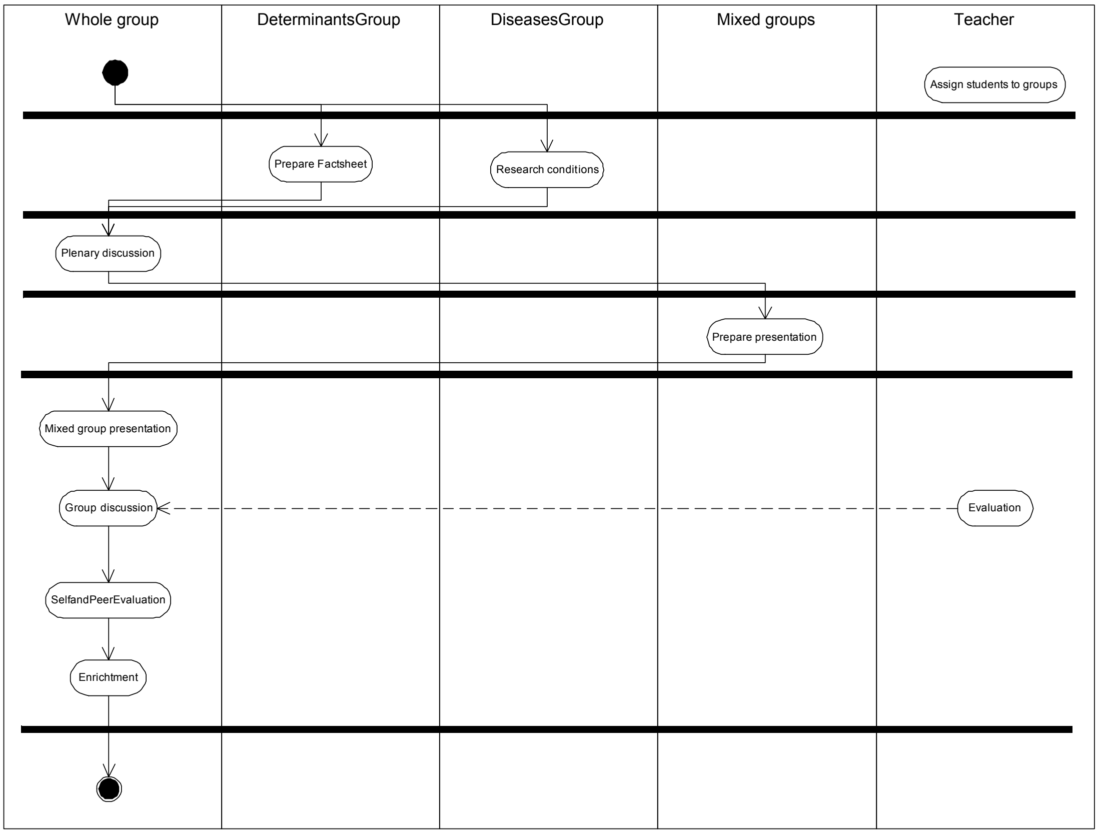


Figure 15. Example of a lesson plan worked out as an UML activity diagram.

A Learning Design instance is then created from the activity description. During the modelling process it is systematically logged where and what kind of difficulties are encountered. Figure 16 shows an example of a lesson plan coded in IMS-Learning Design. An instantiation of the Learning design instance could be created and played in LD compliant player to see the results.

```
- <organizations>
  - <imsld:learning-design identifier="learningdesign1" level="C" uri="http://www.alfanet.portal-ace.com">
      <imsld:title>How environment affects the health of Canadiens</imsld:title>
    - <imsld:components>
      - <imsld:roles>
        + <imsld:learner identifier="WholeGroup" create-new="allowed">
        + <imsld:learner identifier="DeterminantGroup" create-new="allowed">
        + <imsld:learner identifier="DisseasesGroup" create-new="allowed">
        + <imsld:learner identifier="MixedGroups" create-new="allowed">
        + <imsld:staff identifier="Teacher" create-new="allowed">
        </imsld:roles>
      + <imsld:activities>
      + <imsld:environments>
      </imsld:components>
    - <imsld:method>
      - <imsld:play identifier="play1" isvisible="true">
          <imsld:title>Play1</imsld:title>
        - <imsld:act identifier="act1">
            <imsld:title>Act1</imsld:title>
          - <imsld:role-part identifier="Part1a">
              <imsld:title>Part1a</imsld:title>
              <imsld:role-ref ref="DeterminantGroup" />
              <imsld:learning-activity-ref ref="PrepareFactsheet" />
            </imsld:role-part>
          + <imsld:role-part identifier="Part1b">
          </imsld:act>
        + <imsld:act identifier="Act2">
        + <imsld:act identifier="Act3">
        + <imsld:act identifier="Act4">
        </imsld:play>
      </imsld:method>
    </imsld:learning-design>
  </organizations>
+ <resources>
</manifest>
```

Figure 16. Example of a lesson plan coded in IMS-Learning Design.

3.5 Results

3.5.1 Expert analysis

Two experienced learning designers were asked to estimate the difficulty to express a lesson plan in IMS Learning Design on a three-point scale. The estimation options the experts had were (a) no problem , (b) recoverable error, (c) no recoverable error. The initial rating results showed only a slight inter judgement agreement (Cohen's kappa $\kappa < .21$) between the experts. Analysis of the comments the experts provided along with their judgement revealed that one expert estimated all classroom based lesson plans as lesson plans with a recoverable error. If a lesson plan was judged as having a recoverable error based only on a classroom situation then it was recoded as having no problem, because IMS Learning Design is not limited to on-line or distance education.

The inter judgement agreement for the experts was substantial (Cohen's kappa .61 < $\kappa < .8$) after the data was recoded and is shown in table 3. The experts estimated that it would be possible to express all the lesson plans in IMS-Learning Design, the category of “no recoverable error” is therefore not shown in the table.

The experts agreed on 3 of the 5 recoverable errors identified in the lesson plans, each experts found one additional recoverable error on which they did not agree.

Lesson plan number	Expert 1		Expert 2		Expert agreement
	No problem	Recoverable error	No problem	Recoverable error	
1.		x	x		
2.	x		x		x
3.	x		x		x
4.		x		x	x
5.	x		x		x
6.	x		x		x
7.	x		x		x
8.		x		x	x
9.	x		x		x
10.	x		x		x
11.	x		x		x
12.		x		x	x
13.	x		x		x
14.	x		x		x
15.	x			x	x
16.	x		x		
Total	12	4	12	4	14

Table 3: Difficulty to express a lesson plan based upon the expert analysis.

3.5.2 Document analysis

In total 5 recoverable errors were found with the document analysis, non recoverable errors where not found.

	Error type		Instructional stream		
Lesson plan number	No problem	Recoverable error	Empiricist	Rationalist	Pragmatist-sociohistoric
1		x	x		
2		x		x	
3	x		x		
4		x			x
5	x			x	
6	x		x		
7		x			x
8	x				x
9	x		x		
10	x				x
11	x				x
12	x				x
13	x		x		
14	x			x	
15	x			x	
16		x		x	
Total	11	5	5	5	6

Table 4*: Difficulty to express a lesson plan based upon the document analysis and classification of a lesson plan to an instructional stream*

The results of the document analysis are shown in table 4, the non recoverable errors category is not shown.

3.5.3 Pedagogical flexibility

The difficulties to express the lesson plans in LD were categorised per major stream of instructional theories as shown in table 5. These data were not analysed further because the number of observations were too small to obtain sufficient power for statistical tests.

	Error type	
Stream of instructional theory	No problem	Recoverable error
Empiricist	4	1
Rationalist	3	2
Pragmatic-sociohistoric	4	2

Table 5. Difficulties to express lesson plans in LD organised per major stream of instructional theory.

3.5.4 Learning design coding

During the coding of the lesson plans the same difficulties to express a lesson plan in LD were found as during the document analysis as shown in table 6. However, occasionally differences were found with the document analysis but these differences had to do with the interpretation of the lesson plan workflow rather than with the possibility to express something in IMS-Learning Design. These differences were not systematically logged.

	Error type	
Lesson plan number	No problem	Recoverable error
1		x
2		x
3	x	
4		x
5	x	
6	x	
7		x
8	x	
9	x	
10	x	
11	x	
12	x	
13	x	
14	x	
15	x	
16		x
Total	11	5

Table 6. Difficulties found during the lesson plan coding

3.6 Solutions to the identified problems

The results of the test showed that some of the selected lesson plans contained elements for which LD did not provide a standard solution, and an adequate way to describe these cases needed to be searched for. There was no evidence found that LD was not suitable for describing contemporary education, since no situations were found that were impossible to work out using IMS Learning Design. What is of interest is to take closer look at those situations that were not possible to describe directly with LD. Therefore all cases that had a judgement “recoverable error” either in the document analysis or in the expert analysis will be discussed next and a suggestion how to code these cases is given.

3.6.1 Case 1

The first situation that was found dealt with passing a piece of work from one student to another students within a group as illustrated in Figure 17.

IMS Learning Design allows the creation of groups by defining roles. Also, it is possible to create learning objects which are placed in an environment. Furthermore, a person in a role can be notified as soon as person in a role has complete some activity. However, the problem at hand is that it is not possible to let a learning object circulate among other learners within the same role as is the case here.

Students: *Pass your paper* to the person on your right. Write one answer for number (3) for the paper you just received. Your answer must begin with the first sound in the person's name (e.g. Mary - made a mess). Then *pass the paper again* and write an answer for (4), again using the same sound that begins the name. Continue doing this until all the blanks on all the papers are full. You should have lots of different answers from all the people in your group when your paper comes back to you!

Figure 17. Passing on a learning object within a group among all group members

The solution developed for this case uses properties and sub-roles to show or set a property value as illustrated in figure 18 For a group of three learners, three role parts are created. In the first act each learner fills in a field and thereby setting an IMS Learning Design property. Once all learners have completed this activity the next act becomes active. Now each learner sees the property value set by another learner to which the learner now has to respond by filling-in a form and thereby sets a new IMS Learning Design property. If all learners completed this activity, act 3 becomes available. In the last act the learner sees the property value of remaining learner and responds to information filled in by the previous learner.

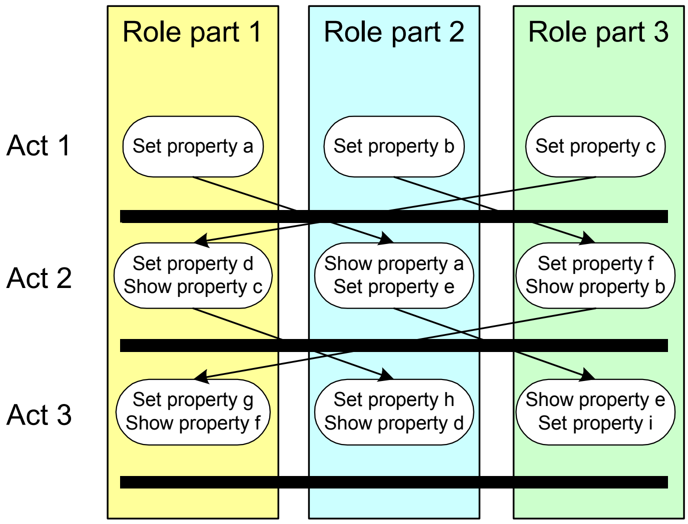


Figure 18. IMS Learning Design implementation for a circulating learning object

The provided solution works fine as long as the number of learners in a group are known beforehand, is fixed, and a group is always complete. Otherwise the workflow needs to be adapted to the different number of learners.

3.6.2 Case 2

The lesson plan where this situation occurred dealt with diving tables that divers need to be able to create when they use compressed air to dive. See figure 19 for the text fragment of the lesson plan that shows the problem. This type of situation could occur also in other situations where safety precautions are needed such as in a construction task or a laboratory experiment.

Important Note
Diving can be a dangerous sport, which is why it's one of the few recreational activities that certifies participants. The Diving Table on page 8 is loosely based on dive tables used by the U.S. Navy without decompression stops and is included here for the purpose of introducing the basic concept of diving physiology. Its utility is limited to this purpose only. Potential divers must receive proper instruction by enrolling in a diver training program offered by recognized certification agencies.

Figure 19. Warning information prior to a learning activity

IMS Learning Design has no special element for this type of information, but there are other ways to reach a similar effect. The easiest way to warn learners for some danger is to include a warning message within an learning activity as instructional text or graphics. An alternative is making use of notifications. As soon as a learner starts with an learning activity that needs a warning message, a property is set *<datetimeactivity-started>* which is compared with the date and time when the activity was published. When the property value that was set is of a later time and or date then the published date a *notification* is send to the learner containing the warning or safety precautions.

3.6.3 Case 3

In two lesson plans a situation occurred where a randomisation mechanism was needed. In one lesson plan students had to draw a piece of paper from a bag (see figure 20) and another lesson plan used randomisation to provide one student from the group with a special task (see figure 21).

> Have one student cut apart Activity Sheet 1 and place the slips of paper in the paper bag.
>
> Group students in pairs. Allow each pair to draw a slip of paper from the bag and discuss the situation described.

Figure 20. Warning information prior to a learning activity

> **A)** Run a lottery to decide who will play the part of the disabled person, small pieces of paper are pulled from a bag and one is marked with a cross.
> **B)** Ask the class to open their papers together. What are their feelings before they open the paper? After finding out whether it is them or not, how do they feel?

Figure 21. Random setting of personal property

In IMS Learning Design there is no mechanism build in that provides randomisation. For the problem of selecting an assignment an alternative using IMS Learning Design might be provided by creating a activity selection where the number of activities is set when the selection is considered completed. That is, if the selection contains 10 activities the learner has to complete only 2 before the whole selection is considered completed. One could also construct a web page (external service) that informs the learner what to do, the learning activity then only contains a link to this web page.

Solving the problem of assigning one learner out of a group of learners with a special characteristic can also be done by IMS Learning Design but not randomly. On this occasion the characteristic did not involve performing different learning activities. Therefore a tutor could set a local-personal property *<locpers-property>*of one of the learners in a role. If a learner has to be assigned a different role, a course administrator has to assign this role to one of the learners and might use the same procedure as described in the lesson plan.

3.6.4 Case 4

On three occasions groups needed to be formed dynamically once a lesson was already started. One lesson plan made use of two types of groups, each containing their own learning activities. At a certain moment new groups needed to be formed based the old groups as shown in figure 22. In principle this means that if there initially two types of groups A and B, at a certain moment new groups are formed out of groups A and B where one member was a member of group A and another was member of group B. Another lesson plan instructed learners to form their own group (see figure 23), which is no problem in class situation but not so apparent using an e-learning platform. The third lesson plan instructed the teacher to divide the whole class into groups as shown in figure 24.

Part A
Divide the class into groups with two or three students in each group.
Half of the groups will be **Determinants Groups** and the other half will be **Diseases Groups**.
Part B
Form new groups connecting the relevant Determinants Group with the corresponding Diseases Group. Each combined group shares Fact Sheet information and prepares an oral presentation for the rest of the class.

Figure 22. Forming new groups out of previous groups

Give students a few days to think about what they will include in the skit and with whom they will work. Let them choose their partners to write and enact a skit that summarizes life in the 1930's.

Figure 23. Formation of groups by learners.

Divide your class into groups, and ask each group to create an aerial map of an area surrounding and including your school (without, of course, using any technology but their own imaginations).

Figure 24. Warning information prior to a learning activity

IMS Learning Design does not provide a mechanism where a learner can assign himself to a group, how learners are assigned to a role depends on the implementation of the run time environment and the administrative system that is used.

Role population during delivery is very similar to the initial role population in the production stage. The main difference is the actor using this functionality. During the production stage role population is considered to be an administrative task, dividing all assigned users of a run to either the staff role or the learner role. This user does not have to have any knowledge of the learning design itself.

During the delivery stage the assignment to roles is further refined depending on the role definitions in the learning design. The users who performs this task needs knowledge of the learning design and, equally important, of the users. For example, if the learning design defines a role of chairperson, knowledge is required about how this role is used in the learning design. The role could be defined to facilitate some kind of learning process of a group of learners. In this example, it is conceivable that a user most capable of fulfilling the facilitator role, is assigned to it in order to optimise the total process. However, if that particular role has been added to the learning design with the objective to improve the skills needed for this type of role, it is more in line with the design's intentions to assign a user to this role who lacks these skills.

Therefore additional information should be included providing instructions for the person(s) that assigns the learners to a new role explaining how the new groups should be formed.

For the example in figure 10 the lesson plan states that students themselves should form new groups. Students can discuss with each other to determine with whom they want to work together and then individually assign oneself to a role. For the examples in figure 9 and 11 the teacher must be able to assign learners to a sub-group. The runtime system should take care of these requirements in order to make these lesson plans work.

The runtime should also provide a mechanism to the user that allows switching of roles. Switching roles implies that the learning design is viewed upon from a different perspective. The user should only be exposed to the role choices that he or she has been assigned to (see role population). LD has a provision for adding information about the role. This information should be presented to the user when the roles are presented. This information informs the user what is expected when assuming this role.

3.6.5 Case 5

Another teaching technique found in one of the lesson plans is often used in workshops and seminars, and provides an overview of what people know prior to the session and what they want to learn during the

> Begin a class discussion by using a KWL chart [what the students know (K), what the students want to learn (W), and what they did learn (L)]. Elicit from the class what they already know about the depression, Roosevelt's New Deal, and the WPA.

Figure 25. Learner inventory form (KWL chart)

session. Afterwards it is evaluated what students actually learned during the session. In this lesson plan this technique was called a KWL chart, see figure 25.

The illustrated problem can be approached in two ways. The first approach uses the conference service as defined in IMS Learning Design and the second approach uses properties and the monitor service of IMS Learning Design. Using the conference service makes it possible to assign different rights to the learners, for example participant, observer, moderator, or a conference-manager. One of the learners or the teacher can be assigned to the role of moderator who collects the responses of the participants to the questions asked to fill the KWL chart and then transfers these responses into IMS Learning Design local properties. Local properties are available to everyone who is subscribed to a run of a unit of learning using the show property value. The second approach uses global personal properties to let every learner fill in a value of the KWL chart on their own. If a monitor service is created then through this monitor the values entered by all learners can be displayed to everyone.

3.6.6 Case 6

While many of the lesson plans investigated held instructions for the teachers on how to use the lesson plan, one lesson plan almost only consisted of instructions and suggestions for teachers.

> Introduction
>
> This unit was developed from the standpoint of a self-contained classroom where the same teacher would deliver the English, Reading and Social Studies instruction. The reading selections, activities and lessons are designed for fourth and fifth grade students, but can be adjusted to meet a variety of reading levels. There is no suggested timeline. This unit can be carried out in its entirety or dispersed throughout the year. It can be integrated with any literature program that is supported by student writing.
>
> ….
>
> The reason I chose memoir writing is because it deals with two difficult issues facing all writers (1) what to include and (2) what not to include. The author, Maya Angelou, once said, "This is a good 20 page paper, if I had had more time it would have been an excellent 10 page paper." In her book, How I Became a Writer, Phyllis Reynolds Naylor shares her view on the evolution of her work, "I've learned to let a manuscript sit for a few days or weeks, then read it again. …

Figure 26. Teacher notes that serve as background information for the teacher who intends to use this lesson plan.

Currently there is no specific activity that covers this need, but there are basically two ways to come close to it. In principle the information stated in figure 26 provides information about a lesson plan and is therefore meta-data. A meta data specification that can be used for this purpose is IMS Metadata for which in IMS Learning Design a name space is provided. In IMS Metadata there is a tag called description in the branch of education which use is to provide comments on the conditions and use of the resource (learning activity in IMS Learning Design). However, there is a limitation of 1000 characters for this field. Another way to provide information to a teacher on how to use a lesson plan is to make use of support activities. Although this type of activity is intended to provide activities that support learners one can also interpret the instructions of the lesson plan creator as support to the teacher who teaches the course. The support activities containing the teacher instructions can be coupled to staff role so that only the teacher has access to these support activities.

3.6.7 Summary

In this test to express a set of lesson plans in LD we found six distinct cases that needed extra attention. The first case described a mechanism for a collaborative assignment that used a circling document among the members of a group of learners. The second case described how a message could be shown before someone started an activity, in this case it was a safety warning. The third case described the use of a randomisation mechanism that was needed select one member of a group. The fourth case identified the need that groups of learners have to be created at runtime. The fifth case described how the pre-knowledge, learning objectives and achieved learning objectives for a group of learners can be captured of each individual learner and exposed to the whole group of learners. The sixth case described the need to capture instructions from the lesson designer or a fellow teacher on how to use a lesson.

3.7 Conclusions

In this evaluation we have taken several lesson plans and investigated how well these lesson plans could be expressed in LD. Although several lesson plans needed a work around, the main educational processes of these lesson plans could be described well with IMS Learning Design. On all but one occasion the work around did not influence the learning process itself but only a small element of it. Only the workaround described in case 1 affected the main learning process. IMS Learning Design offers set of services that have proven to be useful, such as mail, conference and a monitor. However, specific learning situations might need special services which are currently not offered in IMS Learning Design. For this in IMS Learning Design a mechanism is provided that offers the possibility of including services developed elsewhere. For example Hernández, Asensio Pérez, and Dimitriadis (2004) have developed a service specifically for CSCL. We identified the need for two kinds of services in this test, the first one for circulation mechanism of a learning object within a group where each member can edit a part of the learning object, and the second one a random selection of a group member who is assigned a different role. There also exists a need to form new groups at run time based on the outcomes of the learning process. The formation of groups at runtime is something which is foreseen in LD but that depends on the implementation of the runtime environment.

We used three methods to test the expressiveness of LD because we also wanted to gather information on the effectiveness and efficiency of each method. Of the used methods the expert analysis was the most efficient one, the time spend by the two experts was less than time spend on the document analysis and learning design coding. We also experienced that the expert analysis needs to be handled with great care. It is necessary that the experts receive training prior to their rating activities so they interpret and rate situations in the same way. The reliability of the results is expected to increase as more experts rate the lesson plans, but this will be at the cost of time efficiency. It was not difficult to find experienced learning design coders, but it was difficult to find learning design coders that had sufficiently broad experience. The document analysis proved to be more effective and the results are more reliable than those of the expert analysis. We base this on the fact that during the coding of the lesson plans no additional work arounds were identified than already found in the document analysis. However, this method is less efficient since it takes about a factor 3 times as much time than the expert analysis with 2 experts. In this test we only used one person to carry out the document analysis. Persons that do the document analysis need to have the same qualifications as the experts previously mentioned. Finally the learning design coding is the most time consuming method. It takes about a factor 10 the time spend on the document analysis to code a lesson plan in LD. This time could be shortened when specific LD editors become available, for this test we used a generic XML editor.

Future test can make use of this test by further elaborating the used the methods and refine the measurements. The method of document analysis would be the preferred method because it has a good balance between efficiency and effectiveness. Quantitative measures require the analysis of many more lesson plans done in this test. To find out if the pedagogical flexibility requirement is met by LD this type of test is needed and for this one might explore the possibility of a automated document analysis tools.

4. Integration of learning standards

This section provides an overview of the adaptation that has been realized by applying several learning technology standards that each provide limited functionality such that the desired adaptive functionality of the system as a whole could not have been feasible, and make them interoperable.

The used learning technology standards are:

- IMS-Meta Data (IMS-MD)
- IMS-Content Packaging (IMS-CP)
- IMS-Learning Design (IMS-LD)
- IMS-Question and Test Interoperability (IMS-QTI)

These learning technology standards were used to realize:

a. Adapt the course entry point of a learner depending on the learners' pre-knowledge on the subject the course deals with
b. Adapt course content according to the learners' learning characteristics
c. Adapt an assessment in a course part depending on scores realized on self tests throughout the course part
d. Adapt recommendations to learners based on study progress, learning activity, and activities of other learners studying the same subject.

Each of the above mentioned course adaptations needed one or more of the learning technology standards to achieve the desired functionality besides the system modules that handle the complex data streams and interpretation of the learning events.

Next short descriptions will be given on how the course adaptations a through d were realized using what learning technology standards.

4.1 Course entry point adaptation

The objective of this type of adaptation is to customize the course delivery to the needs of an individual learner without having to customize the course design as a whole. This means that a course is designed meeting the needs of generic learner but the moment an individual learner takes the course it is determined what parts of the course are relevant for this learner. One way of using this functionality is to ascertain a learner's knowledge on the course subject and match this with the course objectives of the course. Those parts that the learner already masters can be left out, for learning objectives of which it is not entirely clear that the learner masters these only the assessments are provided to determine if these objectives need to be studied, and learning objectives clearly not mastered by the learner are entirely placed in the course.

For this adaptation the course content should be described with IMS-LD. IMS-LD provides generic functionality that otherwise had to be programmed into the system per individual course. Learning Design offers the possibility to cluster learning activities in structures, thereby creating course modules. Attached to such a module are course objectives, and to be able to measure if a learner has mastered the learning objectives questions are defined per learning objective. In the Learning Design conditions are defined that react on the change of properties, that are also defined in the Learning Design. The properties are defined such that they represent the mastery of a learning objective in the course. When the property value indicates that a learning objective is already mastered by the learner then the condition corresponding with this property changes the visibility of a course module. The result for the learner is that the content corresponding with the mastered learning objectives is omitted from the course. When the property value indicates that there is the possibility exists that the learner has mastered a course objective but not very convincing, than the condition corresponding with this property hides the course content corresponding to this learning objective and only shows the assessment for this learning objective. The idea behind this mechanism is that the pre-knowledge assessment contains a representative sample of the assessment of a learning objective. In other words when there is reasonable doubt whether or not the learner has mastered a learning objective based on the pre-knowledge assessment the whole assessment corresponding to the learning objective is used to ascertain the learner's knowledge on that learning objective. When the property value indicates that the learner has not mastered the learning objective the corresponding condition realizes that both learning content and assessment are shown to the learner.

The questions used in the course should be described with IMS-QTI as well as the assessments. Each test item or question is provided with meta-data (using IMS-MD) that makes it possible to select individual test items for a given learning objective and include this in the pre-knowledge test presented to the learner. An assessment results in an overall score for the whole test and also in scores for the covered learning objectives.

The process is outlined in figure 27 and follows these steps. First, a learner selects the pre-knowledge test from the course, this causes data exchange in the system from the IMS-LD interpreter to the IMS-QTI interpreter to start opening the test window. Second, when the learner starts the pre-knowledge assessment the QTI package is transferred from the Content server to the IMS-QTI interpreter. Third, the assessment results in scores which need to be transferred to the IMS-LD interpreter. The scores are stored in the user portfolio which is stored in the administration module. Fourth, the IMS-LD interpreter is triggered that a change in property is detected in the user's portfolio, these changes are imported by the IMS-LD interpreter and the conditions are evaluated with the new property values. The results are displayed to the learner.

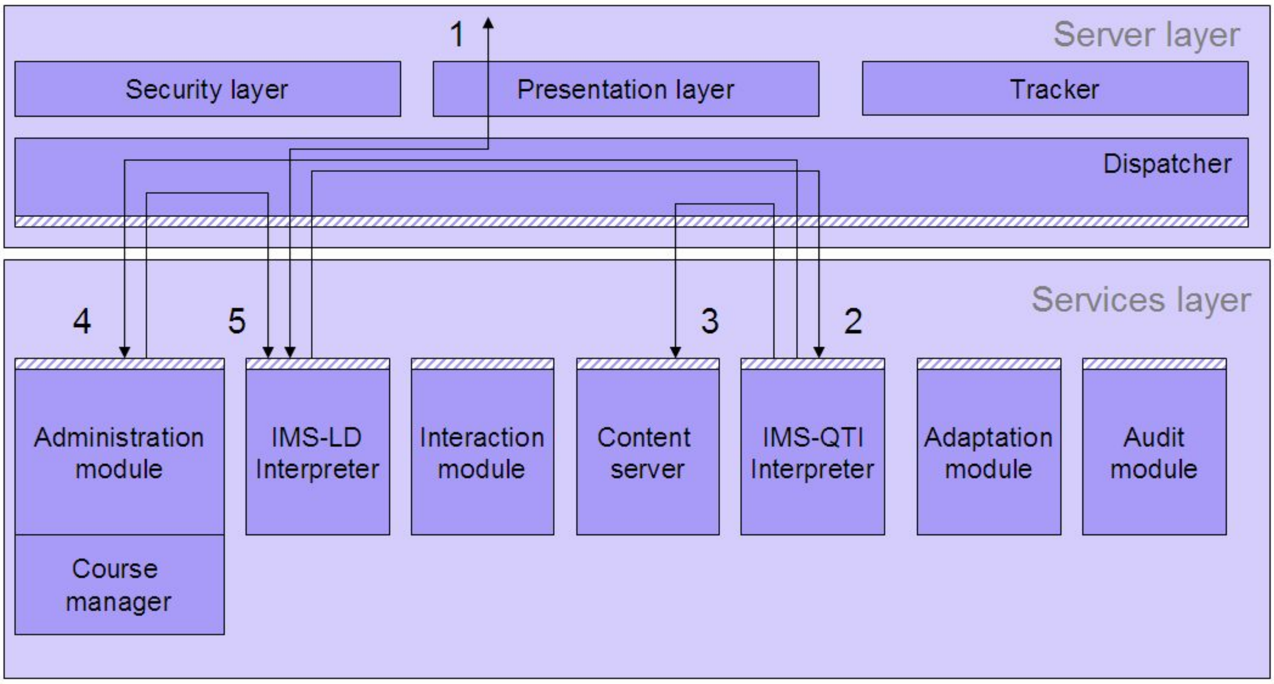


Figure 27. Data flow in the system to process a pre-knowledge assessment.

4.2 Content adaptation

The objective of this adaptation is to facilitate the learner with learning content that is formatted such that it corresponds with the learner's characteristics. Content adaptation takes place on two dimensions; learning style and cognitive modality. Cognitive modality determines if the learner is confronted first with examples then with the rules and vice versa, learning style determines whether the format of the content is visually oriented or verbally oriented.

For this type of adaptation the course should be described with IMS-LD. It requires that more than one version of the content is made available; depending on the dimensions the author wants to include. Dimensions can also be combined in for example inductive/visual and inductive/verbal resulting in an even finer grade of adaptation towards a learner's preferred way of learning. In the learning design properties and conditions are defined. The properties hold the preferences of a learner on the two dimensions; learning style and cognitive modality. The conditions translate these preferences into the selection of only those content blocks that are feasible for that learner. This means that only those content blocks are shown that evaluate true on the properties representing the learning style and cognitive modality, all other content is hidden from the learner.

To determine the learner's learning style and cognitive modality questionnaires are used which should be described with IMS-QTI. The responses to the items in the questionnaire are translated to preferences on one of the two dimensions. The resulting score is stored in the learners' portfolio. When one of these properties of the learners' portfolio is changed then the IMS-LD interpreter is triggered and the new property values are evaluated. The process follows a similar data flow in the system as outlined in figure 1.

4.3 Test adaptation

The objective of this adaptation is to encourage learners to test their knowledge on a regular basis through self-tests that are provided after each course module. The assessment, which is provided after each lesson, is made adaptive to prevent those questions covering modules that were already mastered before as proven by the self-test are repeated in the assessment.

For this type of adaptation is necessary that the self-tests and the assessments are described with IMS-QTI. The items of the self-test and the assessment are provided with meta-data using IMS-MD to indicate the learning objective it covers but also the course module it covers. The self-test results in individual scores for each module it covers and an overall score of the test. These scores are placed in the learners portfolio. When the learner decides to start the assessment, the property values of the self-test that are of relevance for the assessment are read. When the score on a self-test is above a certain threshold, then those questions that address that particular module will be omitted from the assessment. This way only those questions for modules that are not yet mastered will be presented to the learner.

The course should be described with IMS-LD from where the self-test and the assessment are started. In the course properties are defined that contain the scores on lesson modules of the course. When a learner takes a self-test scores are set in the learner's portfolio. Both the self-tests and the assessments are wrapped-up as content packages using IMS-CP.

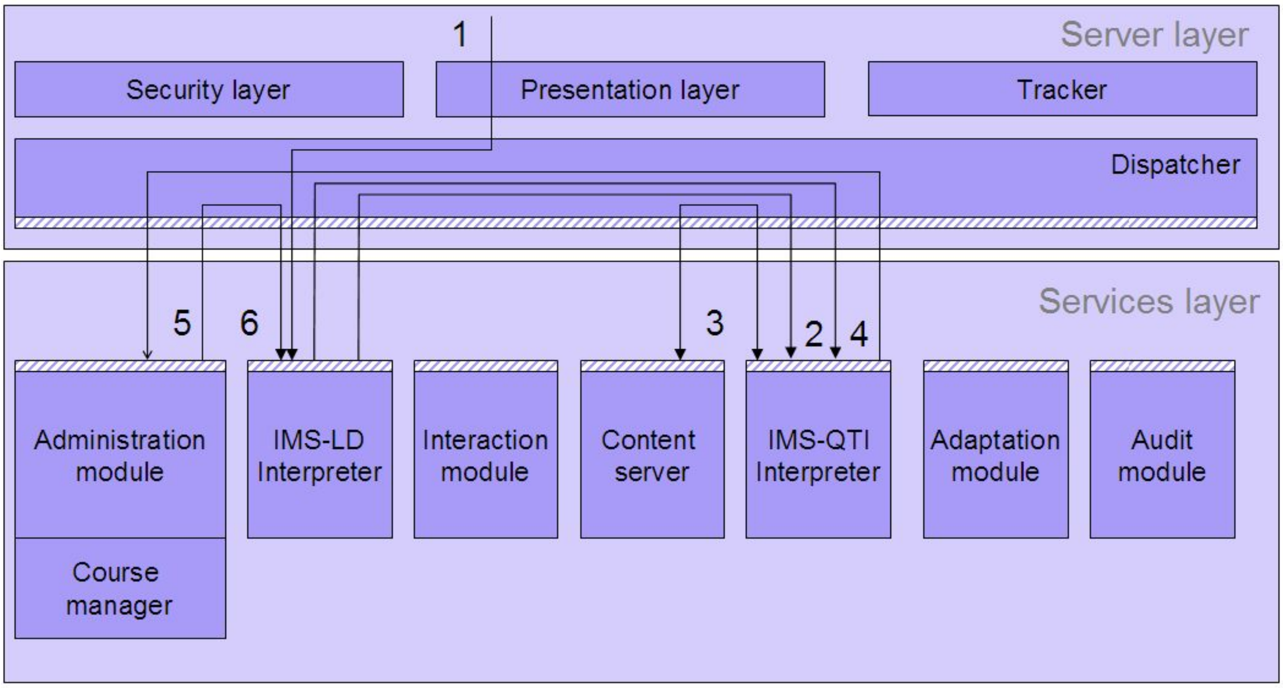


Figure 28. The data flow to create an adaptive test.

The process to adapt a test to previous scores on self-test is outlined in figure 28 and follows the steps described below. We presume that the scores on the self-test are already available, these scores are captured in LD properties. When the learner selects the lesson assessment (step 1) the IMS-LD interpreter triggers the IMS-QTI interpreter (step 2). The IMS-QTI interpreter requests the test content and receives this data (step 3). Step four, the assessment outline is dependent of the self-test scores, these scores are requested from the IMS-LD interpreter. The IMS-QTI interpreter only shows those questions for items which did not score above a threshold during the self-tests. The final score on the assessment is stored in the learner's portfolio (step 5) and this change in value in the learner's portfolio triggers the IMS-LD interpreter to retrieve the updated assessment score (step 6)

4.4 Learner recommendation adaptation

The objective of this adaptation is to guide and motivate learners through a course when they encounter problems related with the course content or when they only partly use the facilities present in the learning environment.

This adaptation requires that the course content is provided with extra information which can be interpreted by the Interaction module. The extra information is described with IMS-MD, and contains descriptions like the learning objectives and the part of the course this material is contributing to. Based on the learner profile that uses the IMS-LIP specification the Adaptation module monitors each individual learner and keeps track of actions taken, contributions provided, and participation in forums. When a learner takes for example a self-test of a lesson and fails this test the adaptation module is able to search for remedial activities for an individual learner based on meta-tagging of the course material and on what other learners have done after failing a that particular test.

Another example of this type of adaptations occurs when a learner is working on a part of a course and progresses at a high pace while at the same time another learner is having difficulties progressing through that part of the course. The adaptation module has the capability to detect that one learner is having difficulties and other learners not, and can generate a message stimulating the learners to take part in a forum and exchange information with each other.

5. Use of user profiles to adapt a course to personal preferences

5.1 Introduction

As learners are actively working on courses provided in the Alfanet system, data needs to be collected, stored and shared of the learner has performed, is working on, and has exchanged with other learners. The adaptive features of the Alfanet system are triggered by the data that is collected on individual learners. This data serves as a base to alter a course for an individual or these data are used to reason upon collective learner behaviour and make alterations in courses for groups of learners. The learner data is stored in a portfolio that is designed according to the IMS-LIP standard. This chapter describes how determining a learners' learning style, cognitive modality and his/her present knowledge on a course specific domain creates the initial learner profile. It is explained how, based on the learners' profile, changes are made to the outline of the course material and what standards are used to realise these adaptations.

When a user fills and submits a questionnaire, the Alfanet core receives an event with all the test results. These results are given by parameters that are described by a name of a variable and its value. Other information about the questionnaire is its identifier and the user that filled it The module in charge of the data model synchronization detects, through the questionnaire identifier format, if it contains information about the user model. Once it is detected that the questionnaire has information about the learner, the module gets all the questionnaire results and, by interpreting the result name, assigns the value to the corresponding learner property.

There are some learner properties that depends on each course objectives. Those properties are the knowledge level and the interest level, and the related objective is extracted also from the result name. So, in order to obtain a right operation from the system, it should be taken into account, for those questionnaires, both the questionnaires identifiers and the assessment variable names.

Finally, and once the properties have been set in the User object, the user profile is updated in the database, in which is stored as a IMS-LIP document.

5.2 Learning styles

Felders's Learning Style Questionnaire, that includes the following attributes: processing (active vs. reflective), perception (sensing vs. intuitive), understanding (sequential vs. global) and sensorial (visual vs. verbal) extended with another attributes that allows to differentiate the type of presentation of the contents (deductive/inductive)

According to Felder[1] learning in a structured educational setting may be thought of as a two-step process involving the reception and processing of information. In the reception step, external information (observable by the senses) and internal information (arising introspectively) become available to students, who select the material they process best and ignore the rest. The processing step may involve simple memorization, inductive or deductive reasoning, reflection or action, and introspection or interaction with others. The outcome is that the material is either "learned" in some sense or not learned. A *learning-style model* classifies students according to where they fit on a number of scales pertaining to the ways they receive and process information.

A student's learning style may be defined in large part by the answers to five questions:

1) What type of information does the student preferentially perceive: sensory (external)—sights, sounds, physical sensations, or intuitive (internal)—possibilities, insights, hunches?

---

[1] Felder R. M., Silverman L. K., 'Learning and Teaching Styles In Engineering Education', *Engr. Education*, 78(7), 674–681 (1988)………….

D. Merrill, Instructional strategies and Learning Styles: Which takes Precedence?, in *Trends and Issues in Instructional Technology*. Reiser *et a*l. (Eds.), Prentice Hall. 2000.

2) Through which sensory channel is external information most effectively perceived: visual—pictures, diagrams, graphs, demonstrations, or auditory, verbal—words, sounds?

3) With which organization of information is the student most comfortable: inductive—facts and observations are given, underlying principles are inferred, or deductive—principles are given, consequences and applications are deduced?

4) How does the student prefer to process information: actively— through engagement in physical activity or discussion, or reflectively— through introspection?

5) How does the student progress toward understanding: sequentially—in continual steps, or globally—in large jumps, holistically?

Currently, ALFANET is focused on two dimensions of the Felders's learning styles: [3] focused on sensorial or cognitive modality (visual/ verbal) and organization (deductive/inductive) attributes. The last one is also called the learning style.

5.2.1 Learning style: Organization: Inductive / Deductive

Induction is a reasoning progression that proceeds from particulars (observations, measurements, data) to generalities (governing rules, laws, theories). Deduction proceeds in the opposite direction. In induction one infers principles; in deduction one deduces consequences.

Induction is the natural human learning style, but Deduction is the most common teaching style: Stating the governing principles and working down to applications is an efficient and elegant way to organize and present material that is already understood.

The learning style questionnaire consists of 5 questions with each two response options, the questionnaire is shown in figure 29. One response option is an indication for the deductive learning style (marked with a D), the other response option is indicative for an inductive learning style (marked with an I).

	1.	How do you like to see presentation of information?
D	a.	Providing a global view before presenting details.
I	b	Providing details before presenting a global view.
	2.	What is the best way of understanding a learning content?
I	a.	Having first examples and then getting the definition of a concept (rule).
D	b.	Having first the definition of a concept (rule) and then getting some examples.
	3.	What is easier for you?
D	a.	Deriving the characteristics of a concept (rule) out of examples.
I	b.	Finding examples of a particular concept.
	4.	What do you find as a more convincing logical construction?
D	a.	"All mice like bier. This is a mouse. Therefore this mouse like bier."
I	b.	"The mice we tested like bier. These mice are typical. Therefore all mice must like bier."
	5.	What do you find more important?
I	a.	Mastering details.
D	b.	Discovering a general rule.

Figure 29. Learning style questionnaire

5.2.2 Cognitive modality: Visual / Verbal

The ways people receive information may be divided into three categories, sometimes referred to as modalities: visual—sights, pictures, diagrams, symbols; auditory or verbal — sounds, words; kinaesthetic—taste, touch, and smell.

Visual learners remember best what they see: pictures, diagrams, flow charts, time lines, films, demonstrations. If something is simply said to them they will probably forget it. *Verbal* learners remember much of what they hear and more of what they hear and then say. They get a lot out of discussion, prefer verbal explanation to visual demonstration, and learn effectively by explaining things to others. Most people of are visual while most college teaching is verbal—the information presented is predominantly auditory (lecturing) or a visual representation of auditory information. Presentations that use both visual and auditory modalities reinforce learning for all students.

The cognitive modality questionnaire consists of 4 questions, the questionnaire is shown in figure 28. For each response to the question it is indicated whether it is an indication for a visualiser or a verbaliser.

1	**When you forget something:**
1.1.	Does it tend to be names, but you remember faces and places? (visualiser)
1.2.	Do you forget faces and places, but remember names and stories you were told? (verbaliser)
2.	**When you learn:**
2.1.	Do you prefer to see text, pictures, diagrams, demonstrations? (visualiser)
2.2.	Do you like to listen to verbal instruction, talks and explanation? (verbaliser)
3.	**Which is easiest for you to concentrate?**
3.1.	On watching something (visualiser)
3.2.	On listening to something (verbaliser)
4.	**When you learn foreign language:**
4.1.	Do you prefer reading written text? (visualiser)
4.2.	Do you prefer listening spoken dialogue? (verbaliser)

Figure 30. Cognitive modality questionnaire

5.3 Creation of a learner profile test

ALFANET computes the User Model (the learner profile is also called the learner model or user model, mainly by the Adaptation Module) using different approaches: with direct questions to the learner, monitoring the users interactions, applying inference rules and using machine learning techniques.

ALFANET provides several forms for gathering the initial learner profile. The questions are organised in four clusters:

- Personal Data
- Habits and Preferences
- General Knowledge

The personal data are mainly used in ALFANET to find similarity between users and research demographic information.

The learner profile consist on the following information:

The *Learner Model* is an extension of the IMS LIP standard, including attributes grouped in personal data, learning styles, background knowledge, habits, preferences and interests.

The *Interaction Model* is commonly considered as a part of the User Model, but we separate it from our Learner Model because of its level of complexity and its relevance for our study. This interaction model describes the learner experience and progress in a global context: Current Situation, Statistics and Historical interaction in ALFANET.

At course level the following interactions are stored:

- *Interaction items* (II) are generated by the learner during the course experience as result of the interaction within the Services. The scope can be individual or shared with the rest of learners. Examples of II are: Messages in a Forum service, files or URL's in the File Storage service.
- *Characterization Items* are interactions that characterizes, gives the opinion of the learner about other LI (provided by the author or other learner). Examples are users' comments and ratings.
- *Learning Items* (LI) are both II and the Learning Objects provided by the author at design time.
- Interaction Events are passive accesses to a LI.
- *Evaluation Item* provides a result of the interaction with a Learning Object of type IMS QTI[2] (questionnaire or evaluation of an activity).
- *Organizational Item* is used to organize (grouping, classifying) the interaction items. Examples are Threads in a Forum service, folders in the File Storage service.
- *Categorization Event*: The learner can assign LI to one or more categories, facilitating the conceptual understanding.
- *Link Event* defines a relation between two LI.

For each interaction we analyse the time spent, the size and the amount of interaction produced.

Other information stored is:

- *Recommendations* provided to users.
- *Sessions* that usually reflect the concept of a running class.
- *Navigation path*: this is the most basic information that can be used to infer any other relation between different items at run-time.

Based on the experience, some *individual interaction* and *collaborative interaction* indicators have been identified to be relevant for the e-learning process. These indicators will be obtained from the interactions performed by the learner, using inference and machine learning. Some of the identified relevant indicators are:

- *Participative* student (impulsive or selective): produces lots, and useful, contributions
- *Insightful* student: is able to detect from the beginning the other users contributions that are most valuable
- *Useful* student: participates in the contributions that are highest evaluated
- *Non-collaborative* student: behaves as if there is no collaboration among students
- Student with *Initiative*: performs new tasks
- *Skilled* student: has appropriate knowledge to do the tasks in the system
- *Communicative* student: has the ability to transmit information to other group members
- *Level of support* (helps to other students)
- *Reputation*: considers the quality in the collaboration interactions with the system and the other students, such as the knowledge of the student, the ability to transmit it and the utility for the rest of students

[TBC] Relevant characteristics obtained and used from Habits& Preferences and General know quest.

5.4 Use of learner profiles to determine the learners entry position

In language courses for beginners you always face the situation that there are real beginners without any knowledge of the language they intend to learn as well as people with some – buried - pre-knowledge, so called false beginners. In most cases the false learners do not know at what level they should start at a given course and here a pre-knowledge test is helpful for them.

[2] IMS Question & Test Interoperability
http://www.imsglobal.org/question/index.cfm

The KLETT reference course consists of 4 lessons. Each lesson consists of several modules like grammar (GR), listening and reading comprehension (RC), and communicative ability (CA). The hierarchy of the lessons is fixed with an increase of difficulty (difficulty levels being very easy for lesson 1; easy for lesson 2, medium for lesson 3 , and high for lesson 4). Every lesson makes use of knowledge and structures of the previous lesson(s) and adding new content for each of the modules. Before going on to the next higher lesson the learner has to do a final lesson assessment for the lesson s/he is actually working at, in order to make sure that s/he is fine with the contents of this lesson.

The pre-knowledge test for false beginners makes use of these final lesson assessments: From every module of every lesson one or two representative test items are chosen to be used for the pre-knowledge test. We have divided the pre-knowledge test into two parts: part 1 assessing the knowledge regarding lesson 1 and lesson 2, part 2 assessing the knowledge provided in lessons 3 and 4. For each part we assemble a questionnaire: Pre-knowledge test part 1 containing the questionnaire with the chosen items from lesson 1 and lesson 2; pre-knowledge test part 2 containing the questionnaire with the chosen items from lesson 3 and lesson 4. The learner starts with the questionnaire of part 1 and according to the results s/he gets the recommendation where to begin the course.

6. Adaptive testing

6.1 Introduction

To determine if a learner makes progress in mastering the learning material, the learner has to take formal assessments during the course, but a learner can also periodically test his/her learning progress. In the Alfanet system tests are designed using the IMS-QTI standard, and two adaptation mechanisms were created that make use of the outcome of tests. This chapter describes how question items are created and how adaptive tests can be created out of these individual test items with additional meta-data and specifically created tool. This chapter further describes how the test results are handled by the Alfanet system to feed them back into the Learning Design of a course. This chapter concludes with an example showing how the score on a test can be used to provide remediation feedback to a learner

6.2 QTI items

IMS Question & Test Interoperability defines an Item as the smallest exchangeable object within QTI-XML. It could be compared with a question but really it is more than a 'Question' in that it contains the 'Question', the presentation/rendering instructions, the response processing to be applied to the participant's response(s), the feedback that may be presented (including hints and solutions) and the meta-data describing the Item.

All information associated to an item provides a detailed report of what to ask, the way to present the questions and the reasons. One of the key responsibilities of the author is to consider these kind of issues during design stage. He/she has to determine the response-type and to map this to the appropriate rendering type. This mapping will depend upon the educational objective of the Item. Similarly, the Section and/or Item groupings, selection and ordering will be dependent upon the educational objectives of the ASI unit.

Regarding to adaptation issues, meta-data generation associated to QTI items plays a very important role. These meta-data fields are used for typifying an item. In design time, author could characterize an item, for instance as more appropriate for inductive learners, or for covering a particular learning objective, or any property from user model.

Meta-data definition and adequate selection rules will be useful for selecting items with characteristics in concordance with the properties associated to user model in a concrete period of time during course execution.

Examples of meta-data that is included in items:

- Learning Objective. Items will be designed for covering a series of learning objectives.
- Knowledge Level. Author designs items for several knowledge levels. This is the way for measuring the mastery level of learners. The Knowledge Level of a learner changes according to course execution, so assessments must be adapted to learner knowledge level in that moment.
- Sub Knowledge level

  It could be useful in order to provide more information about knowledge level of learner. For instance in a remediation scenario, this metadata is useful for generating assessments adapted to "real" learner knowledge level. It intends to express until what extent a learner is getting the concepts of a concrete lesson (i.e. while the knowledge level evolves with the lessons, the sub-knowledge level indicates the knowledge level for a lesson –related with difficulty). Thus, in the Remediation scenario this attribute need to be defined to describe different test items for the same objective but with different difficulty level.

- Purpose. An item could be used for measuring level of mastery of a learner or for practicing an acquired knowledge. Items from one or other purpose could be different.
- Rendering type. It could be interesting to have the possibility of generate assessments using specific rendering type
- Individual traits (features that together define a user as an individual, and have to be extracted by specially designed psychological tests), for instance cognitive factors, learning styles, etc.

6.3 Questionnaires and adaptive tests

Adaptation may be supported according to different user characteristics. Usually there are many features of the current context that may be taken into consideration while the student is working with the system. Such features include characteristics of the user as a person or characteristics of how he/she works. These features could be identified (among others) as student goals, student knowledge, student qualification (how quickly he or she acquires knowledge), experience in the domain, personal preferences, etc.

These characteristics must be taken into account during course design stage. Related to questions and test design, these features must be present too. Authors have to provide materials according to potential audience, and from then on a Dynamic Adaptive Assessment generation assume importance.

QTI Items defined by authors are bundled to create a collection of data objects. These data objects are named object banks in ASI Information Model. An Object Bank is a collection of Items, Sections or a mixture of Items and Sections and it is used to contain the database of evaluation objects that can be used to construct Assessments. The Dynamic Adaptive Assessment used in Alfanet system is based on pre-defined rules for the selection of the relevant items available in existing item banks and characterised with metadata.

The idea of Dynamic Adaptive Assessments is to generate in run time questionnaires according learner characteristics. It recognises the changes in its status and generate assessments according to it. Defined rules in course design time will be in charge by selecting what questions will be part of presented questionnaires to the user. In these rules, authors could define the maximum number of questions that appear in each assessment and in which order. In run time rules defined are fed by user characteristics, taking into account that they will be varied over course execution because of learner evolution.

6.4 QTI and Learning design integration

IMS Learning Design and IMS QTI are natural partners in the learning process. The primary motivation for integrating IMS Learning Design and QTI stems from use cases which involve exploiting the results of a test or assessment to influence the learning process, often referred to as formative assessment.

The integration of IMS LD and IMS QTI revolves around aligning property and variable names. Essentially, when property identifiers and variable names are declared to be lexically identical at design time (i.e. in IMS LD-based and IMS QTI-based XML), they are considered to be a *shared variable* in run-time software environments which involve IMS LD and IMS QTI-based processes.

QTI assessment process is in charge of to evaluate an exam and to generate a score value (or several score values) according to item definition. QTI process has no information to determinate if an assessment is failed or not. Information about required score for passing an exam lies in LD design. From adaptation point of view, is very useful to know in which materials the learner has weakness, and to recommend additional materials in order to overcome such weakness.

To achieve this effect is necessary to generate several scoring variables in item definition time, and in LD definition to manage these variables. I.e.: in order to be able to know in which module the learner fails, it is necessary to have in separate variables score by module (i.e. a scoring variable for items defined for covering Grammar materials, etc), and in turn LD definition must be able to determine if the learner has suitable level of mastery or not.

6.5 Use of QTI to create a remediation scenario

Taking advantage of IMS Learning Design and QTI integration it is possible to exploit the results of a test or assessment to influence the learning process. QTI questionnaires are very useful in order to measure the mastery level of learner and his/her academic skill. Depending on results derived from an assessment, two types of actions are possible. If the test outcomes are favourable, learner gets additional material. If a learner does not reach the required level, then remediation option are suggested.

In a remediation scenario a learner is induced to insist in which materials the learner has weakness, and to recommend additional materials in order to overcome such weakness. At the end of that retrieval process a Remediation assessment could be proposed. Remediation assessment is a questionnaire which is dynamically generated and that makes emphasis on those items in which the user has the worst punctuation. This Remediation assessment is defined from an object bank of test items that incorporate different metadata information as described before, and with the conditions and appropriate selection rules. In a remediation scenario, metadata information is useful for generating assessments adapted to "real" learner knowledge level. It intends to express until what extent a learner is getting the concepts of a concrete lesson (i.e. while the knowledge level evolves with the lessons, the sub-knowledge level indicates the knowledge level for a lesson –related with difficulty). Thus, in the Remediation scenario this attribute need to be defined to describe different test items for the same objective but with different difficulty level.

7. Adaptation based upon interaction of learners with the LMS

This chapter describes how learners can be guided through a course based upon the run-time interactions of learners (the target learner and the group of learners belonging to the course).

As already mentioned, in ALFANET authors are allowed to specify (IMS-LD) mandatory, reinforcement and extension learning objects to be provided to learners at run time. Usually, the reinforcement and extension learning objects are very low specified, and learners have freedom to choose them. Our proposal consists of recommending to each individual learner in each relevant pedagogical situation of the course the most appropriate material taking into account the interactions performed by a group of similar learners. ALFANET can support learners in two scenarios, lack of knowledge and high interest.

7.1 Introduction

When the learner works in a course, the ALFANET system gives her/him some advises related to their learning process, in a similar way that could do a dedicated tutor that is co-operating with the learner. Two type of advises are provided: recommendations about what to do next and informative and motivational messages.

At the design time the author characterises learning objects according with the LOM educational attributes, extended with usage (optional, mandatory, reinforcement). Learning Objectives are also associated to the Learning Objects (let be by direct definition in the metadata catalogue entry).

During the publication of the course, Alfanet system constructs the Course Model, mainly oriented to the Learning Objects, Services and their relation with the course objectives.

The author, the tutor or the administrator can complement the definition of the course by configuring some additional messages or recommendations for the course (reference to manual).

At any time when the learner enters into the system, s/he can fill in the Profiles questionnaires. ALFANET system prepares the User Model (the global data associated to the user with independence of courses).

The tutor can configure a work-group in order to realise a collaborative activity. Thus, the tutor can create new services (forums, folders), as well as a sub-group with access only for the group members.

7.2 Monitoring user behaviour

When the learner enters into the course, fill-in the interest questionnaire. During the learning process the learners works on some individual activities, reading of material, navigating by the objectives description, the activities of the course, also works on different self-assessments and lesson assessments. The learner share their experience with other learners making use of the educational services (forums and file storage) (reference to collaborative scenario).

The ALFANET system completes the user model with data about the interactions of the user in the course (interest and achievement level on each learning objective, number of accesses, comments, value of ratings to each learning object and service). These data are obtained by the Tracker and the Interaction Module and the events generated by the LD Interpreter, the QTI Interpreter and the data are processed by the Adaptation Module in order to model the user interaction.

Using the inference rules that Alfanet incorporates, new user characteristics are computed. Machine Learning algorithms are used to compute the value of some users characteristics mainly considering past interactions already performed in the course by this learner and other learners (level of activity in objective, and in the course).

Based on the dynamic user model (continuously updated with the new interactions performed by the learner), ALFANET diagnosis the current situation of the learner:

- lack-of knowledge in objective
- high-interest in objective

7.3 Guiding learners through a course

The system generates recommendations and messages that are offered to the learner:

When the learner works in the course, the ALFANET system diagnoses a lack-of-knowledge in the working objective, and thus, offers a recommendation for doing the learning material of reinforcement for the target objective.

A learner has worked during a lot of interactions but the system doesn't knows the achievement level for the working objective (and QTI exists). Then the system recommends to do the self-assessment associated to the target objective.

During the course interaction a learner is diagnosed high-interest in one objective with a good mastering in the self-assessment. The system recommends learning material of additional info for the target objective.

The learner can follow the recommendation or ignore it. In some cases a Feedback button is offered to the learner in order to inform to the system about the satisfaction level with the recommendation.

The ALFANET system monitors the progress of the recommendations in order to self-regulate the recommendations in the future. (this feature is pending of implementation, by now, only stores the feedback).

7.4 Requirements for monitoring user behaviour

Requirements IMS-LD for IM-AM.

1. Identifiers for courses and environments: (alphanumerical: only lower case letters and numbers, not ñ, nor special chars as -).
2. Identifiers for courses and environments can't exist in another published course. Recommendation: to name courses and environments with course version i.e. coursev1 envv1, (search and replace v1 by v2 in order to change manually all the identifiers).
3. Definition of services: <service class="FO" > for forums, <service class="FS" > for file-storage. Identifiers of services must be part of a valid URL (not containing special chars, only allowed _). Participants,
4. Objectives defined in the <learning-objectives> as items (identifiers) and associated to the LO within the LOM metadata (catalog-entry=objective).
5. Each objective has one QTI that computes the KL of the objective. This LO is characterised by:
   a) It has the objective associated
   b) Learning resource type is Self-assessment
6. The LO are characterised as mandatory, optional or reinforcement reinforcements
7. URL of the resource for the description of activities and course objectives:
   a) Course: Desc_<course_id).htm
   b. Activity: Desc_<activity_id).htm

8. Safeguard the quality of a course

8.1 Introduction

When courses are designed, the authors have certain expectations with regard to how learners will react to that course. For example, learners are expected to complete a course within a certain amount of time, or authors design a route and learners are expected to follow this route, but are they? To answer questions like these the Alfanet system is equipped with audit functionality that monitors learner behaviour and compares this behaviour with predefined expectations by authors. The results are reported back to the authors with the expectation that when a bottleneck is found in a course it can be put right. This chapter explains the full audit cycle that starts with the design of a course and ends with the generation of audit reports; it emphasizes the use of standards that are needed to collect the necessary data.

8.2 Setting triggers in the learning material using IMS-LD

The development of the Audit function of Alfanet is based upon a life-cycle view of the development and use of a e-learning course.

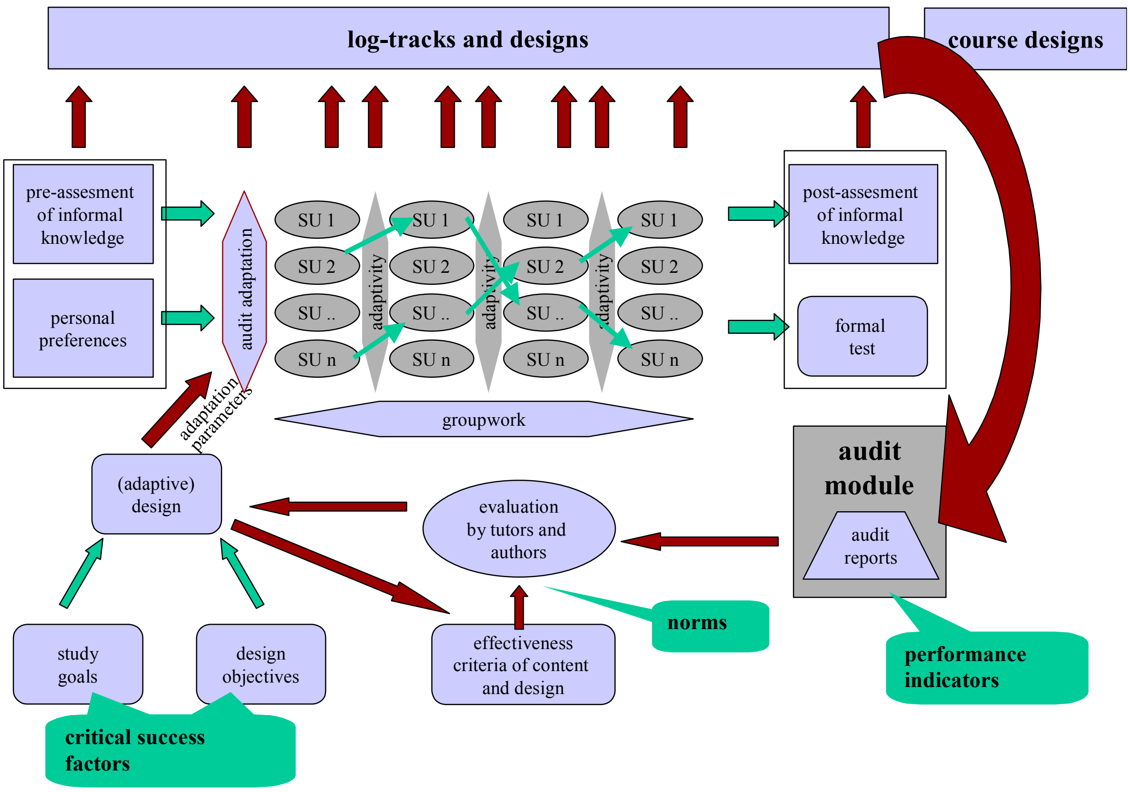


Figure 31. Life-cycle view on a e-learning course.

Central in this view is a development process of course materials which starts with the specification of a number of critical success factors (CSF). The CSF's reflect the high-level objectives of the organisation with respect to a given course.

From these CSF's norms are derived which can be to performance indicators from the running e-learning system.

Examples of CSF's and related norms:

CSF:	a course is used successfully by a majority of learners
related norm:	70 % of learners enrolled use the course
	75% of the learners that use the course are successful
related performance indicators:	45 % of learners enrolled use the course
	95% of the learners that use the course are successful

To be able to perform automatic analysis, the design norms have to be stored in the Learning Design, both in the actual data and the meta-data. Actual meta-data that are currently used are the anticipated study time of an activity and the difficulty-level.

The norms then can be related to the outcomes of the tracker-functions in the e-learning system. In the Alfanet project three standards have been chosen to define the content and the didactics of the course: IMS-LD, IEEE-LOM and IMS-QTI.

When the audit module is activated by a user, the course designs that are stored in IMS-LD are analysed to compare anticipated use of the course with the actual use. The main aspects that are monitored in the Alfanet system are the use of a module, the duration of the use and the order in which the modules are used. By implementing norms in the design, the norms that can be followed when the system is used can be extended at choice. The LOM meta-data that are used in the current design are from the educational category of LOM meta-data; especially the anticipated learning time and the difficulty level are relevant.

The actual use of the course is derived from a tracking module which tracks events like starting and finishing a module and results of formal or informal QTI-tests. The tracker delivers the performance indicators.

It is especially relevant for the Alfanet project, that the audit module can distinguish between anticipated use of the separate modules in a course, and the actual use. Of course this difference between design and actual use is strongly influenced by the freedom that can be given in the design using these standards and the adaptation functions that are explicitly integrated in the design of the project.

The audit module produces a number of standard reports that can be used by course designers to evaluate the design and the outcomes of the study process for a group of learners. There is no automatic analysis and comparison of the norms against the performance indicators, but from the reports the relations can be derived. The data have to be interpreted by a human being who can compare the outcomes with norms and CSF's.

The audit module is not meant to deliver data for a single learner. Because of the chosen high-level feedback-approach, the audit module only produces relevant data for a group of learners.

8.2.1 Examples of reports and data used

Basic report: use of the modules and degree of completion per user

Tracker data used:	information about all the study-events that are related to this course like: learners, activities, QTI's, status with respect to activities and QTI's

OVERVIEW OF THE ACTIVITIES FOR A GIVEN COURSE

PARAMETERS:

CourseId
commtechinformaticav8

RESULTS:

CourseID	LearnerID	ActivityId	ItemType	FirstTime	LastTime	CompleteTime	AccessCount	Value	Score test
commtechinformaticav8	ounl1	learnerProfileTest	activity-structure	9-maart-05 23:41	9-maart-05 23:41	9-maart-05 23:41	1	complete	
commtechinformaticav8	ounl1	sync_qtiresult_LOb1_Toets_SCORE	integer	9-maart-05 23:44	9-maart-05 23:44		1		
commtechinformaticav8	ounl1	LOb1_Inleiding	learning-activity	9-maart-05 23:45	10-maart-05 0:13	10-maart-05 0:13	1	complete	
commtechinformaticav8	ounl1	LOb0_Cognitief	learning-activity	10-maart-05 8:01	10-maart-05 8:03	10-maart-05 8:03	1	complete	13
commtechinformaticav8	ounl1	LOb0_Leerstijl	learning-activity	10-maart-05 8:03	10-maart-05 8:04	10-maart-05 8:05	1	complete	20
commtechinformaticav8	ounl1	LOb1_IntWWW	learning-activity	10-maart-05 8:05	10-maart-05 8:06	10-maart-05 8:06	3	complete	
commtechinformaticav8	ounl1	LOb1_TekstTest	learning-activity	10-maart-05 8:07	10-maart-05 8:08	10-maart-05 8:08	2	complete	
commtechinformaticav8	ounl1	LOb1_Toets	learning-activity	9-maart-05 23:42	10-maart-05	9-maart-05 23:44	1	complete	7

List of Learners that have started a course

Tracker data used: information about the learners that have started this course

LEARNERS THAT HAVE DONE (PARTS) OF A COURSE

PARAMETERS:

CourseId
commtechinformaticav4

RESULTS:

Learners
john
ounl1

Number of study hours for a given course, a learner and an activity

Design-data used: LD meta-data: difficulty-level of the activity

Tracker data used: information about the actual study-time and test results for a given activity and a given learner

NUMBER OF STUDY HOURS FOR THE COMBINATION OF PARAMETERS

PARAMETERS:

CourseId	LearnerId	ActivityId
commtechinformaticav8	ounl1	LOb2_Toets

RESULTS:

Number of study hours	Difficulty	Score test
0.4	Medium	3

Mean number of study hours over all learners for an activity

Design-data used LD meta-data: difficulty-level of the activity

Tracker data used information about the mean actual study-time for a given activity and course activity

MEAN STUDY TIME OF THE ACTIVITY

PARAMETERS:

ActivityId
LOb2_Toets

RESULTS:

Mean study hours	Difficulty
0.3	easy

Difference between anticipated study time and actual study time over all learners

Design-data used: LD meta-data: anticipated study time

Tracker data used: information about the mean actual study-time for a given activity and course

MEAN DIFFERENCE BETWEEN PLANNED AND ACTUAL STUDY TIME FOR THE LEARNING OBJECT

PARAMETERS:

ActivityId
LOb2_Toets

RESULTS:

Mean number of hours
-0.2

List of all activities for a given course.

Tracker data used information about all the study-events that are related to this course like activities and number of accesses

LIST OF ALL ACTIVITIES / LEARNING OBJECTS USED AND NUMBER OF ACCESSES FOR A GIVEN COURSE

PARAMETERS:

CourseId
commtechinformaticav8

RESULTS:

ACtivityId	Nnumber of accesses
LOb2_Inleiding	0
LOb2_GelProt	0
LOb3_AS_ENV_LO_RemTest	20
LOb2_TekstTest	0
LOb3_AS_ENV_LO_RemTestTekst	20
LOb2_Toets	0
LOb3_VoorkennisTest	0
LOb3_AS_ENV_LO_ConceptTest	7
LOb3_AS_ENV_LO_VerbalMaterials	5
LOb3_AS_ENV_LO_VisualMaterialsSinus	5

8.3 Observing learner behaviour in the Alfanet System

The behaviour of the learners are tracked by the tracker module. The tracker-database is queried using XPATH statements; an XML-related standard to select nodes from XML-documents.

The tracker records contain detailed information about the use of the modules, the sequence in which the modules are used and the results of the tests.

9. Learning standards integrated: a working example

9.1 Introduction

In previous chapters different aspects of the Alfanet system have been discussed individually. However, the main thrust of the Alfanet system, compared to other initiatives, is that in this system multiple standards have been integrated to realise a set of adaptive features that should both help individual learners as well as groups of learners to collaborate. This chapter provides the description of two pilot courses in which all functionality of the Alfanet system is integrated which will be shown.

This chapter is organised as follows: First we will outline the set of pedagogical models that were applied at two of the Alfanet pilot sites. These models provide practical guidelines but need implementation specific for each situation. Therefore the translation of the models to the pilot situations is explained step by step. The first step in the design process is the creation of an course outline using IMS-LD. For the design we follow the workflow imposed by the Alfanet LD authoring tool. After the course outline has been created the adaptation rules have to be defined, therefore first properties have to be defined and than the conditions can be composed. Then it is explained how alternative paths can be designed based on learner characteristics. To be able to advise learners at runtime the course material has to be prepared such that it can be interpreted by the adaptation module, it is outlined what needs to be added to the course material. Finally it is explained how adaptive test can be constructed. The tests used in the Klett pilot are used to outline what the principles of adaptive testing require in practice.

9.2 Course design using pedagogical models

The results of research in learning and instruction were captured in pedagogical models. A pedagogical model is defined as a method that prescribes how a class of learners can achieve a class of learning objectives in a certain context and knowledge domain. Pedagogical models are inspired by theories on learning and instruction. Examples of methods are learning Spanish as a second language, how to learn mathematical skills for engineering, or how to plaid in someone's defence during a trial. By defining a template course designers are helped implementing a specific type of instruction such as problem-based learning.

A pedagogical model for concept learning was developed and applied in a language course. The model describes the learning process at two layers, the first layer describes the course outline and the characteristics of the lessons within the course, the second layer describes the learning path within a lesson.

The course consists of various lessons that gradually become more difficult for learners. The target audience for the course are beginners that have no prior knowledge for the language to be learned, but it is also foreseen learners with some prior experience will take the course with the goal to improve their language skills. To support both types of learners the outline of the course adaptation is designed as shown in figure 32. Learners get the opportunity to take a pre-knowledge test; the result of this test is used to determine the optimal starting point for a learner.

A learners' score on the pre-knowledge test is compared with some thresholds that are set for each line in figure 32. The thick black line is the learning path for learners with no or very little prior knowledge, they start with the concept learning of lesson a and when finished the test of lesson a can be taken. When the test of lesson a is accomplished with a sufficiently high score the learner can continue with lesson b and so forth. Learners that have reached a score on the pre-knowledge test above a certain threshold can obtain a different starting point in the course. For example if the relative score on the pre-knowledge test was 45% and this corresponds with the threshold set for lesson c than the learner may start directly with the test of lesson c. When this test is failed, the learner has to start with the concepts of lesson c, when succeeded the test the learner can continue with the next lesson.

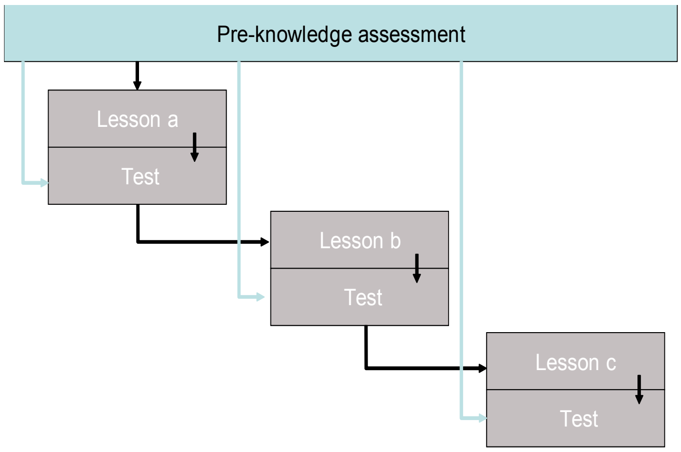


Figure 32. A model to adapt the course content to the learners pre-knowledge.

Within a lesson the learner studies the learning material for the concept that is explained in the lesson. Each lesson contains four variants of the learning material that explains the concept. Each variant is designed according to a learning style combination inductive/deductive and visual/verbal. When the learner thinks that the concept is mastered, a self test can be taken in the practice activity. Based on the test score the learner can proceed to the next concept module or if the test scores are below a predefined threshold a remediation activity is presented to the learner. When finished the remediation, the learner can do the test again, when failed for the second time the learner is advised to contact the tutor for further help, else the leaner can proceed to the next concept module. The flow within a lesson is shown in figure 34.

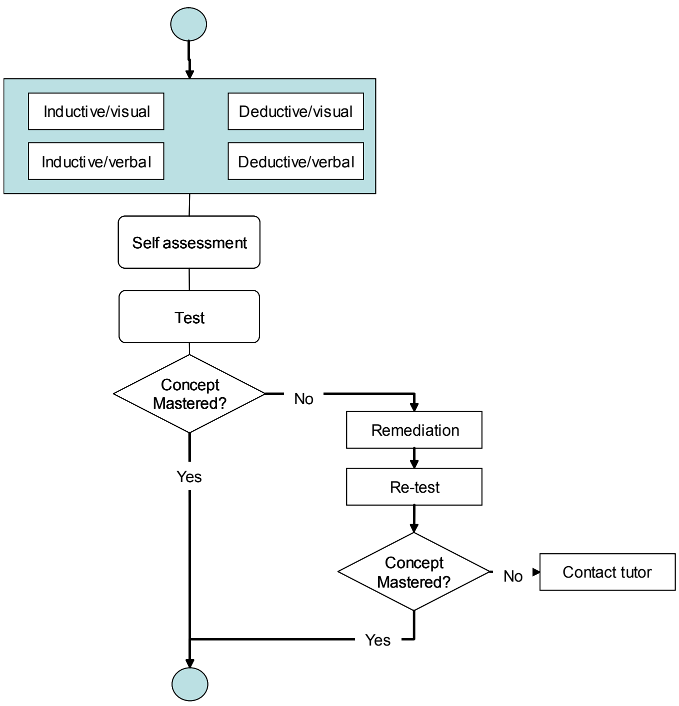


Figure 33. The pedagogical model for concept learning.

Within this template several triggers are included to adapt the flow through the model. First, this model incorporates some of the earlier explained learning styles and adds to that rules that define the learning path through the learning material. Second, the question items presented in the re-test depend on the scores obtained in the previous test. If the score on a test item was above a certain threshold than that item is taken out of the re-test.

9.2.1 Application of the template in a language course

KLETT language course defines 4 Modules: Grammar, Communication Ability, Listening and Reading Comprehension, and Intercultural Competency. When these modules are applied to the pedagogical model outlined in figure 2, the Inductive and visual part of the course looks like figure 3.

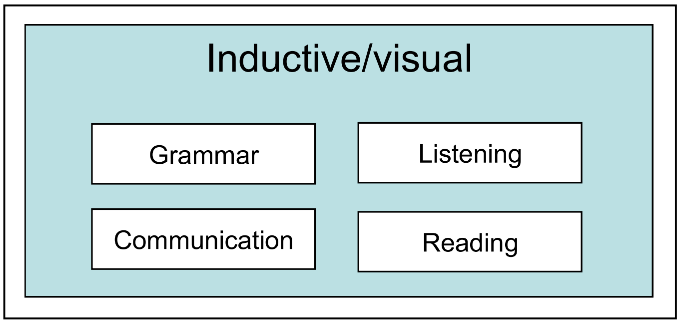


Figure 34. The pedagogical model applied to one of the lessons the Klett language course

The concepts for every module are defined according to knowledge levels, e.g. for the module Grammar the concepts for lesson 1 (low knowledge level) are: nouns with different gender and the respective articles, forms of the verb “ser” in present tense and the subject pronoun. In lesson 2 (medium knowledge level) the concepts of Grammar are: plural of nouns and articles, the form of adjectives corresponds to the related nouns, forms of regular verbs ending on –ar in present tense, how to form negation.

Regarding the presentation of the concept learning content varies according to the learning style and cognitive modality of the learner: there are different exercises for inductive visual learners and for deductive verbal learners.

Practice: The course provides materials for individual self learning, but also activities for collaborative learning are offered.

Examples for feedback provided are e.g. recommendation to do self assessment after having finished a module, when assessments have been failed the learner receives recommendation to do some remediation activities and later to do the remediation assessment

9.2.2 Creating the course structure

To create the structure of the course using the Alfanet editor we have to follow a bottom-up approach meaning that first the learning activities have to be defined, activities can then be clustered into activity structures, and lessons can be created by creating clusters including the previously defined clusters for each module.

9.2.2.1 Creating learning activities

To create a learning activity implies that the learning object to be used by the learning activity is already defined. The same applies to the environment(s) attached to a learning activity, first the learning objects have to be defined, these learning objects can be placed into an environment, then the environment (with the learning object) can be attached to a learning activity.

9.2.2.1.1 Define a learning object

The definition of a learning object involves these steps:

a) From the sub-menu “learning materials” learning objects have to be selected

b) A new learning object has to be chosen

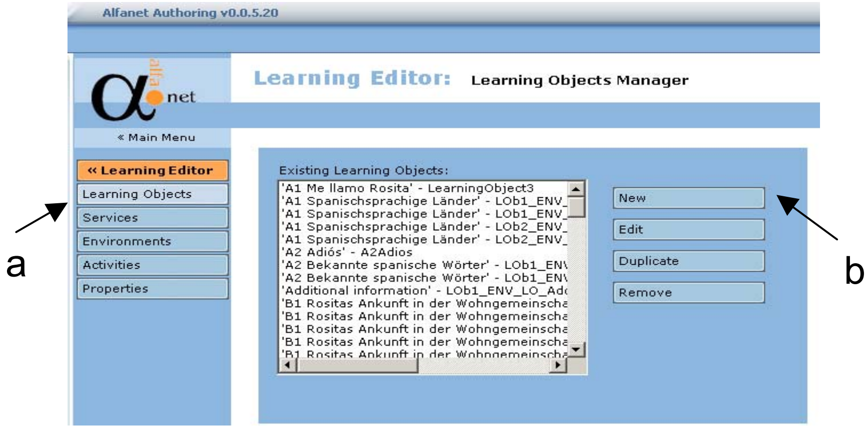


Figure 35. Create a new learning object

A new window opens and three steps have to be followed to define a new learning object:

a) A title has to typed

b) The items tab has to be chosen

c) A new item has to be created

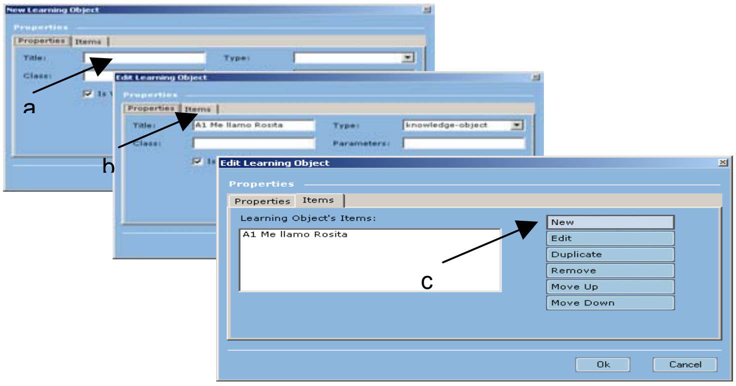


Figure 36. Create a new learning object item.

For the new learning item

a) A title has to be given

b) A new resource has to be created

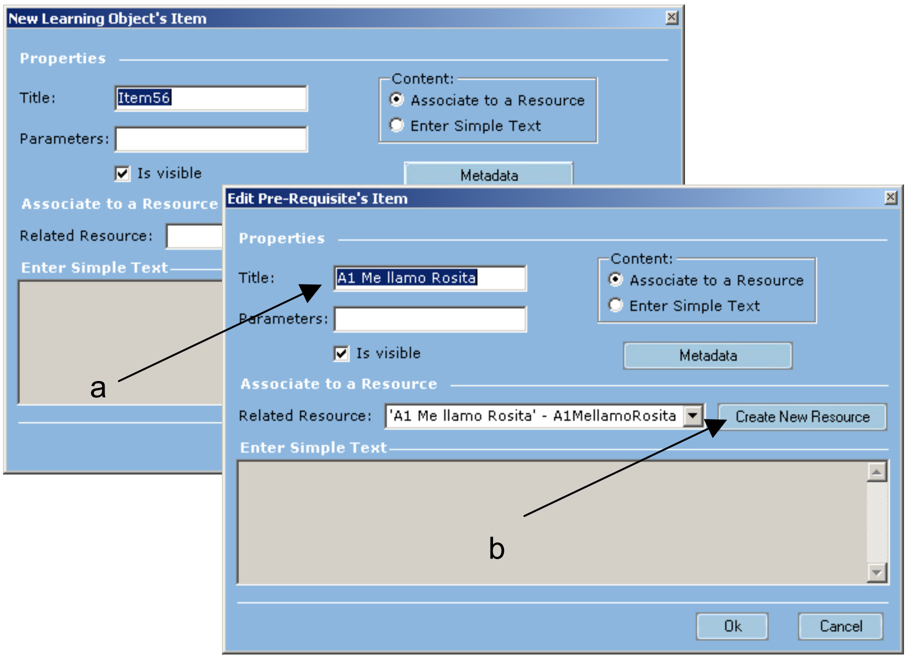


Figure 37. Creating a new learning resource

The new learning resource is created in 4 steps:

a) A title has to be given

b) The creation mode has to be defined

c) The place where the file is located has to be given

d) The location of the associated files has to be given

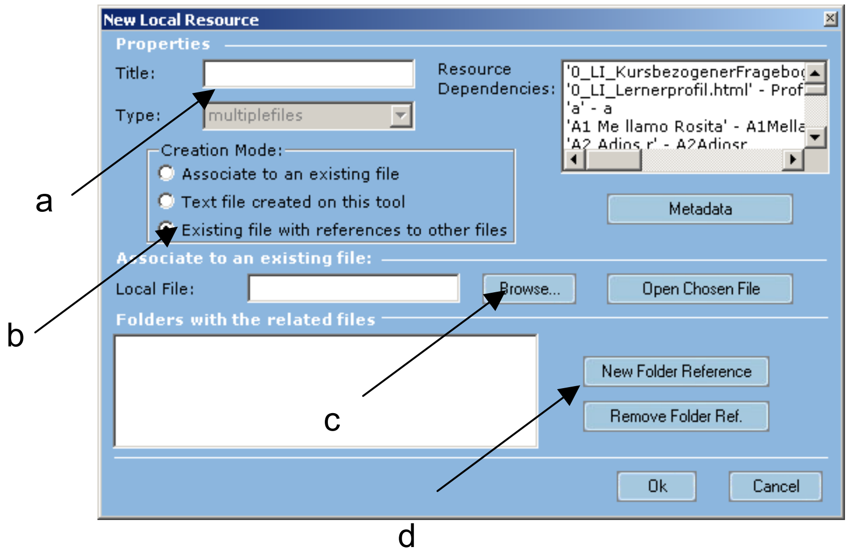


Figure 38. Defining a new local resource.

9.2.2.1.2 Creating an Environment

Learning environments also use learning objects and the same procedure can be followed as outlined above. Once a learning object is defined it can be attached to an environment. These steps have to be followed to create an environment:

a) From the materials sub-menu choose 'environments'

b) Select new

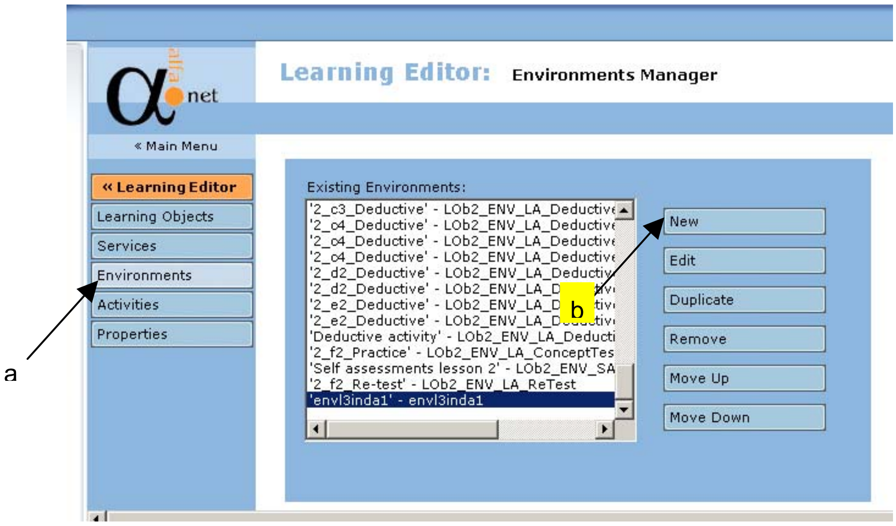


Figure 39. Creating a new environment

The new learning environment is created by:

a) Providing a name
b) Select the learning objects tab
c) Select a learning object
d) Add more learning objects by holding the CTRL key and select multiple objects

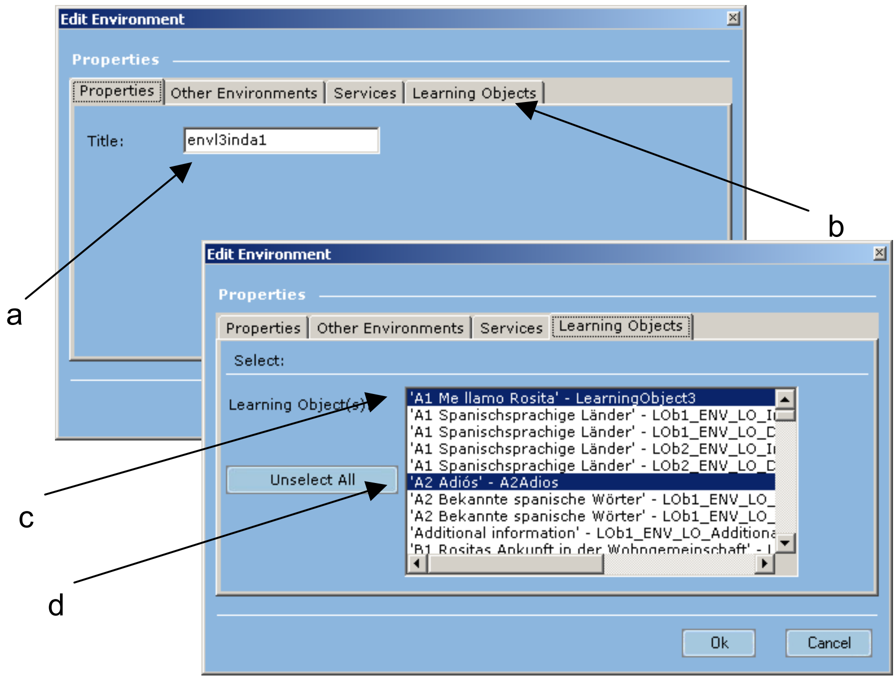


Figure 40. Attach learning objects to an environment.

9.2.2.1.3 Creating a new learning activity

Once the learning objects, and environment are defined a learning activity can be created following these steps:

a) From the learning material sub-menu choose 'learning activity'
b) Choose new

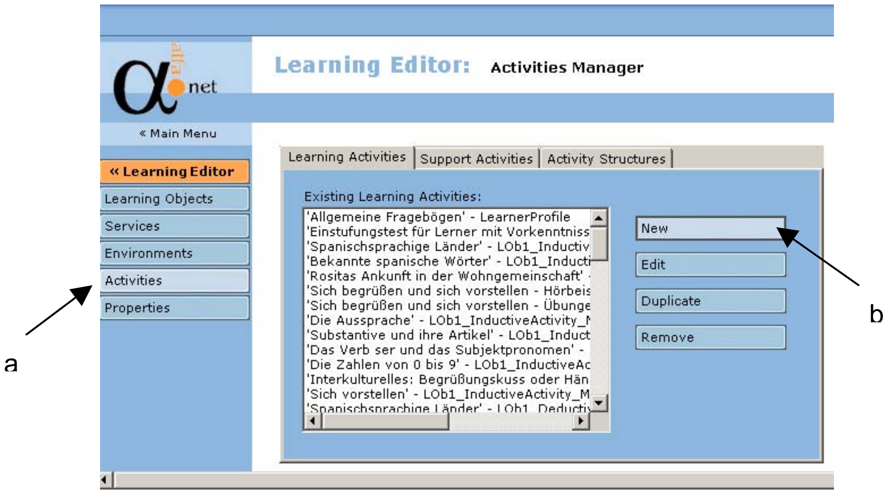


Figure 41. Creating a new learning activity

For the new learning activity:

a) A title has to be entered

b) Choose the tab 'description' and choose new

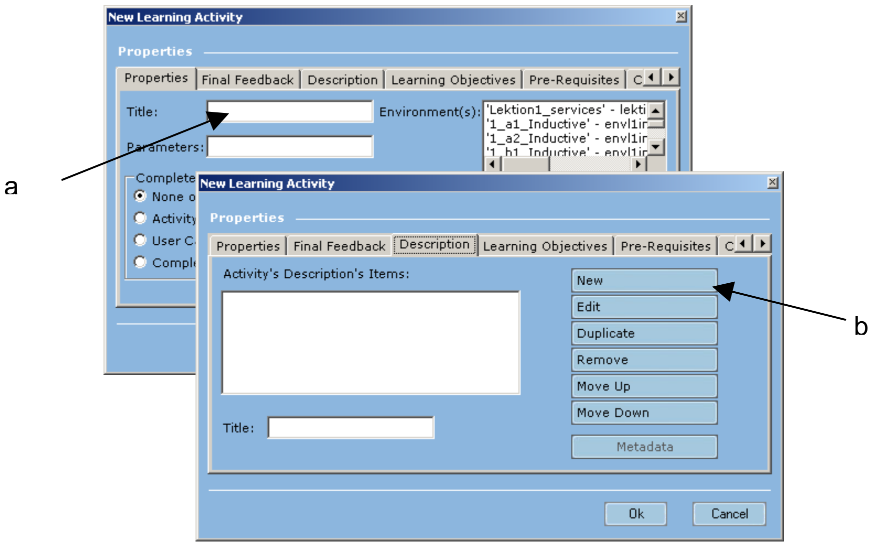


Figure 42. New learning activity description.

The new learning activity description requires:

a) A title

b) Either a simple text for the learning activity or reference to a learning object

c) Selection of an environment if needed.

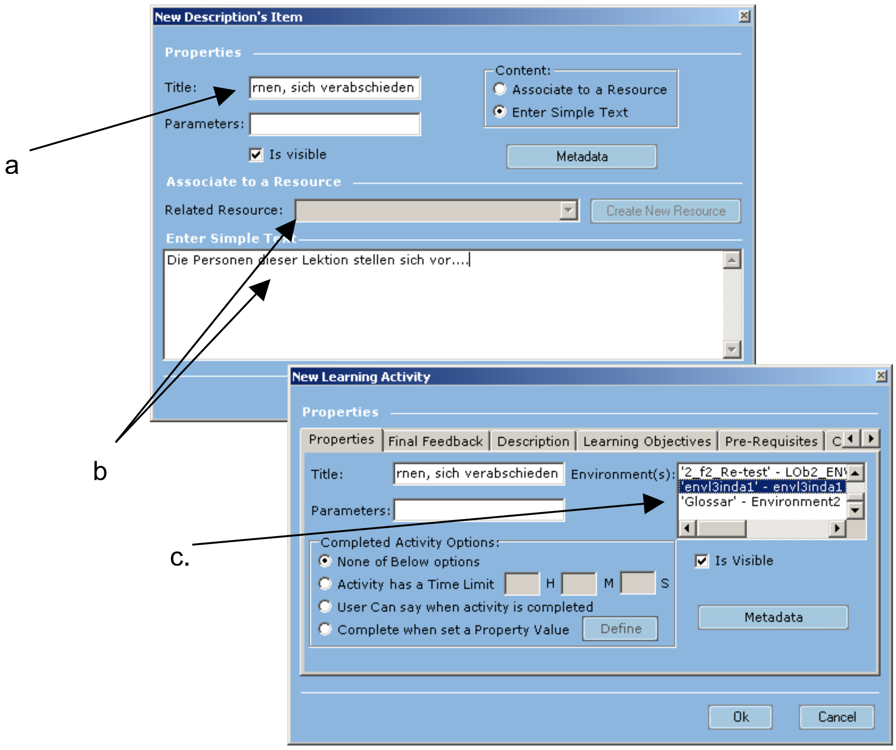


Figure 43. Creating a new learning activity

9.2.2.2 Creating lesson modules

To create a lesson module according to the pedagogical model template, the individual learning activities have to be created as explained above. For each lesson a total of four modules have to be created; grammar, reading, writing, and communicating.

A lesson module is created by:

a) From the learning material sub-menu select Activities and then the tab Activity Structures

b) Select new

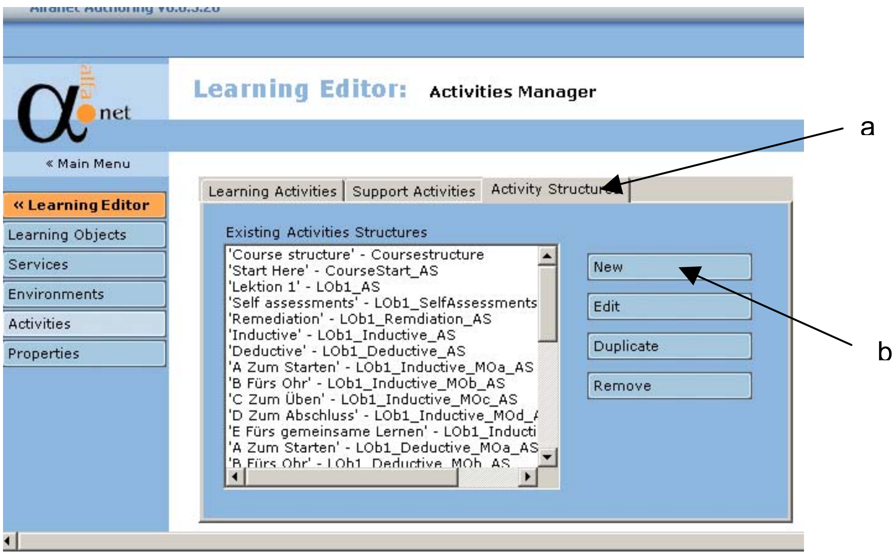


Figure 44. Creating a new activity structure

The new learning activity requires:

a) A title

b) Select the tab Learning activities

c) Select the activities to be included in the structure, and change the order if needed

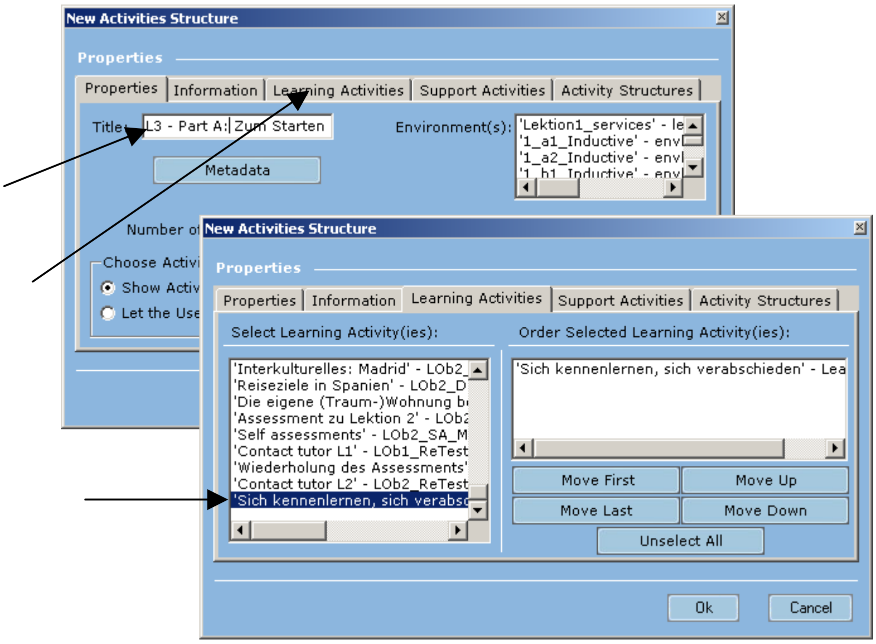


Figure 45. Selecting activities for the activity structure.

9.2.2.3 Creating lessons

To assemble a lesson containing the activity structures defined for the modules these steps have to be carried out (see figure 13):

a) Select from the learning material sub-menu, Activities and then the tab Activity Structures

b) Choose new

For the new Activity structure:

a) Enter a title

b) Select the Activity Structure tab

c) Select the activity structures to be included in the lesson

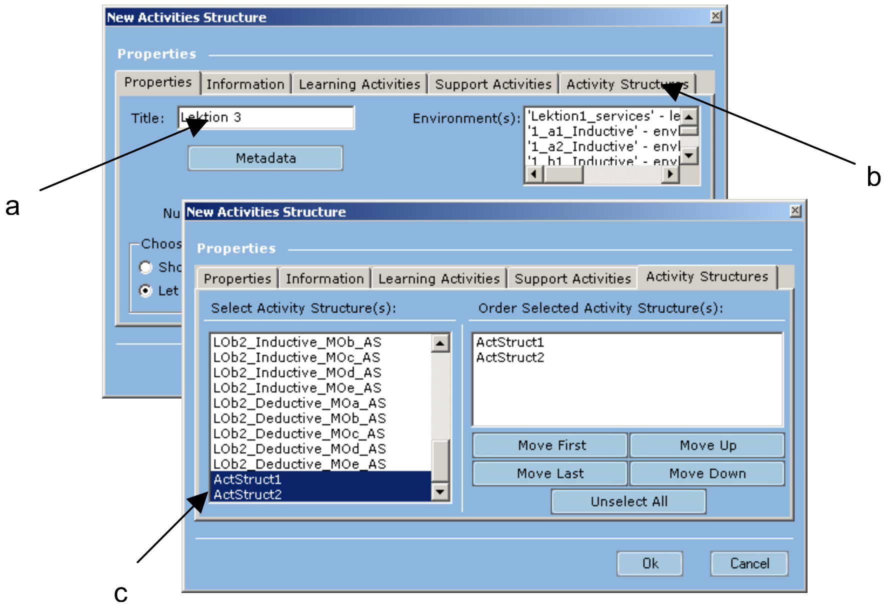


Figure 46. Create a new activity structure containing activity structures.

9.2.3 Defining the Ld adaptation rules

For the IMS-LD adaptation rules to work it is necessary that properties are defined. Then when the properties are created, conditions can be created that manipulate the course structure based on the values the defined properties possess.

9.2.3.1 Defining properties

To create properties that are used locally in the course described for all users these steps have to be followed:

a) Select the properties from the learning material sub-menu.

b) Select the Course Properties tab and select the tab Local in Course and choose New

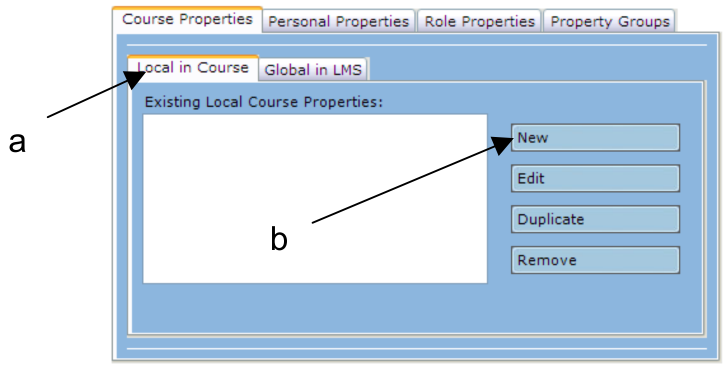


Figure 47. Create a new local IMS-LD property.

The define the new property these data have to be provided:

a) A title

b) Optional the initial value of the property

c) The data type of the property value (i.e. text, Boolean, or integer)

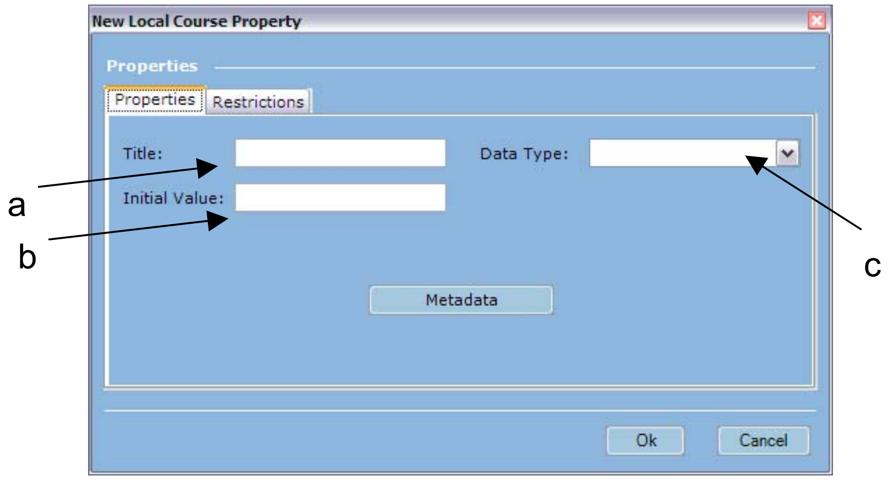


Figure 48. Entering data for a local property.

9.2.3.2 Defining conditions

The IMS-LD conditions can be created once the properties that will be used to define the status of the condition are created as explained above.

To create a condition these steps have to be followed:

a) Select adaptation from the course orchestration sub-menu.

b) Select new from the pre-conditions tab

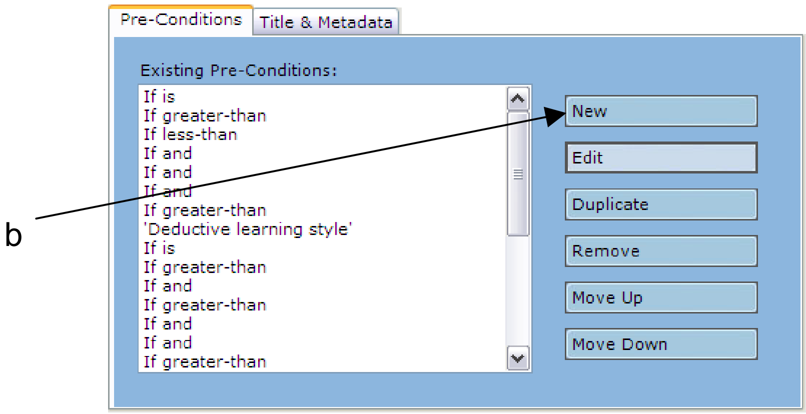


Figure 49. Creating a new condition.

The first part of a condition is the IF statement which require these steps:

a) Select the If tab of the condition

b) Enter a title

c) Choose the expression type

d) Enter the first operand by choosing a property

e) Enter the second operand by entering the property value for which this rule should execute

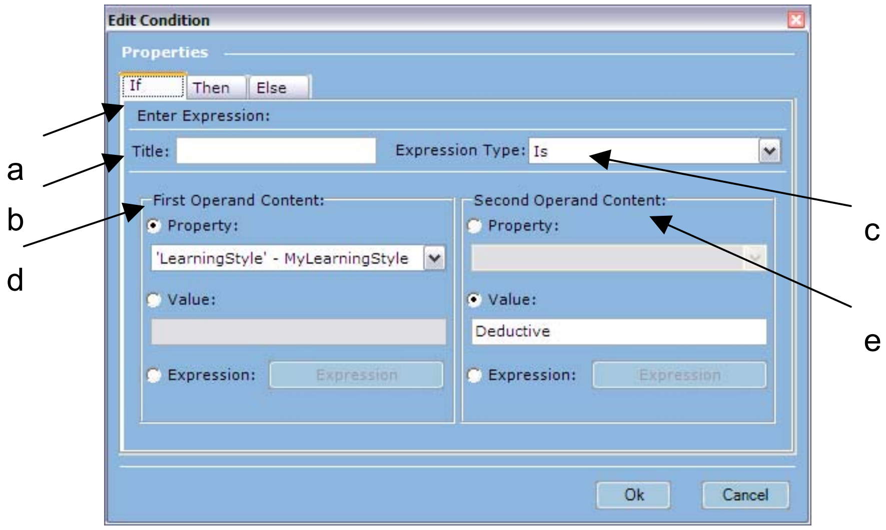


Figure 50. The If part of a condition.

The Then part of a condition defines the action that should be taken if the If part of the expression evaluates true. The then part is defined in these two steps:

a) Select the action that should happen

b) Select the object on which the actions should have effect

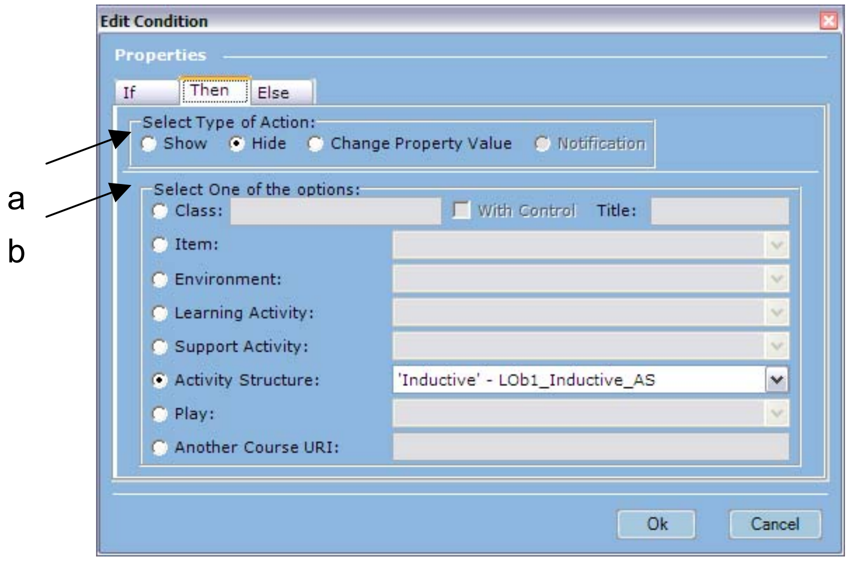


Figure 51. Defining the Then part of a condition.

9.3 A Learner profile resulting in alternative learning paths

In the pedagogical meta-model alternative learning paths are defined according the learners learning style and cognitive modality. To realise this adaptation first the conditions to capture the student's characteristics have to be defined. Then the conditions can be created that use these properties. The procedure to create these elements are described in part 9.1.

Two global personal properties have to be created:

a) Student learning style

b) Student cognitive modality

imsld:globpers-property (2)

	[redacted]	[redacted]				
1	StudentLearningStyle	imsld:global-definition				
			uri	templatev36		
			imsld:title	LearningStyle		
			imsld:datatype			
				datatype	string	
			imsld:initial-value	Inductive		
			imsld:restriction (2)			
				[redacted]		
				1	enumeration	Inductive
				2	enumeration	Deductive
2	StudentCognitiveModality	imsld:global-definition				
			uri	templatev36		
			imsld:title	CognitiveModality		
			imsld:datatype			
				datatype	string	
			imsld:initial-value	Verbal		
			imsld:restriction (2)			
				[redacted]		
				1	enumeration	Verbal
				2	enumeration	Visual

Table 7. Definition of learner profile properties.

Two conditions have to be created that complement each other. One when the learning style is inductive and another one when the learning style is deductive. In table 2 the definition of the conditions are given. Here learning style is used in combination with knowledge level of the student.

From the condition definition it becomes clear that where in one condition an activity is shown, this activity is hidden in the other one, vice versa. Instead of showing or hiding an individual activity also an activity structure can be used.

Comment	rule b shows only the inductive activity if the student profile is set with value and the student is a beginner				
imsld:if	imsld:and	imsld:is (2)		imsld:property-ref	imsld:property-...
			1	imsld:property-ref: ref = KnowledgeLevel	Beginner
			2	imsld:property-ref: ref = StudentLearningStyle	Inductive
imsld:then	imsld:show	imsld:learning-activity-ref	ref	LOb1_InductiveActivity	
	imsld:hide	imsld:learning-activity-ref	ref	LOb1_DeductiveActivity	
Comment	rule c shows only the deductive activity if the student profile is set with value and the student is a beginner				
imsld:if	imsld:and	imsld:is (2)		imsld:property-ref	imsld:property-...
			1	imsld:property-ref: ref = KnowledgeLevel	Beginner
			2	imsld:property-ref: ref = StudentLearningStyle	Deductive
imsld:then	imsld:hide	imsld:learning-activity-ref	ref	LOb1_InductiveActivity	
	imsld:show	imsld:learning-activity-ref	ref	LOb1_DeductiveActivity	

Table 8. Definition of the conditions for the student's learning style.

9.4 Advising learners at runtime

Learner behaviour is monitored at runtime with the goal to provide them with advice on specific learning objects that are present in the course and that are advisable to look at in a specific situation. The adaptation module takes care of this part of user monitoring and advising learning objects. This module is not able to interpret the content of learning objects and therefore additional meta-data has to be added to each learning object.

Learning Objects (LO) are the educational material that traditionally are used in the courses, e.g. books in paper, electronic external material, textual or multimedia digital material. The attributes used in ALFANET are upon the general and educational categories. Also technical attributes are defined.

- Learning resource type (Theoretical, Practical, Assessment).

  Theoretical: *diagram, figure, graph, index, slide, table, narrative text, lecture*

  Practical: *exercise, simulation, experiment, problem statement*

  Assessment: *questionnaire, exam, self assessment*

- Interactivity level (five degrees from low to high: *very low, low, medium, high, very high*).

- Difficulty (five degrees from easy to difficult: *very easy, easy, medium, difficult, very difficult*).

Related categories.

Three different levels of learning objects are going to be considered to provide adaptations:

- Level 1 (L1) mandatory learning objects that have to be read by the learners to achieve the objective.
- Level 2 (L2) reinforcement learning objects for those learners that do not achieve the objective.
- Level 3 (L3) extension learning objects, for those learners that are very interested in the objective.

Other key attributes to consider when provide adaptation were identified:

- Appropriateness -for learners with Learning Style (inductive/deductive)
- Appropriateness -for learners with Cognitive Modality (visual/verbal)
- Appropriateness -for learners with knowledge Level (beginner/advanced)

There are different options to include additional metadata:

1. As keywords (associating pre-defined words to an element). We can include a lot of different keywords; but not grouped values for a concept are expected.

```
<imsmd:keyword>
  <imsmd:langstring>mandatory</imsmd:langstring>
</imsmd:keyword>
```

2. As entry in catalog (is the case for associating objectives). Multiple entry in catalog allowed.

```
<imsmd:catalogentry>
  <imsmd:catalog>objective</imsmd:catalog>
    <imsmd:entry>
      <imsmd:langstring>LObBuscador</imsmd:langstring>
    </imsmd:entry>
</imsmd:catalogentry>
```

3. As educational metadata:

```
<imsmd:educational>
  <imsmd:learningresourcetype>
    <imsmd:source>
      <imsmd:langstring>ALFANET Authoring Tool</imsmd:langstring>
    </imsmd:source>
  <imsmd:value>
     <imsmd:langstring>Narrative Text</imsmd:langstring>
  </imsmd:value>
  </imsmd:learningresourcetype>
  <imsmd:interactivitylevel>
  <imsmd:source>
    <imsmd:langstring>ALFANET Authoring Tool</imsmd:langstring>
    </imsmd:source>
  <imsmd:value>
    <imsmd:langstring>Low</imsmd:langstring>
    </imsmd:value>
    </imsmd:interactivitylevel>
  <imsmd:difficulty>
  <imsmd:source>
    <imsmd:langstring>ALFANET Authoring Tool</imsmd:langstring>
    </imsmd:source>
  <imsmd:value>
    <imsmd:langstring>Easy</imsmd:langstring>
    </imsmd:value>
    </imsmd:difficulty>
  <imsmd:typicallearningtime>
```

```
<imsmd:datetime>00:00:15</imsmd:datetime>
</imsmd:typicallearningtime>
</imsmd:educational>
```

4. As technical meta-data:

```
<imsmd:technical>
  <imsmd:format>text/html</imsmd:format>
<imsmd:duration>
  <imsmd:datetime>00:01:30</imsmd:datetime>
    </imsmd:duration>
    </imsmd:technical>
```

The Alfanet authoring tool supports course designers in adding the needed meta-data for a learning object by using global meta-data which is applicable for all objects in the course design and by providing a meta-data entry screen for each learning object as is illustrated by figure 21

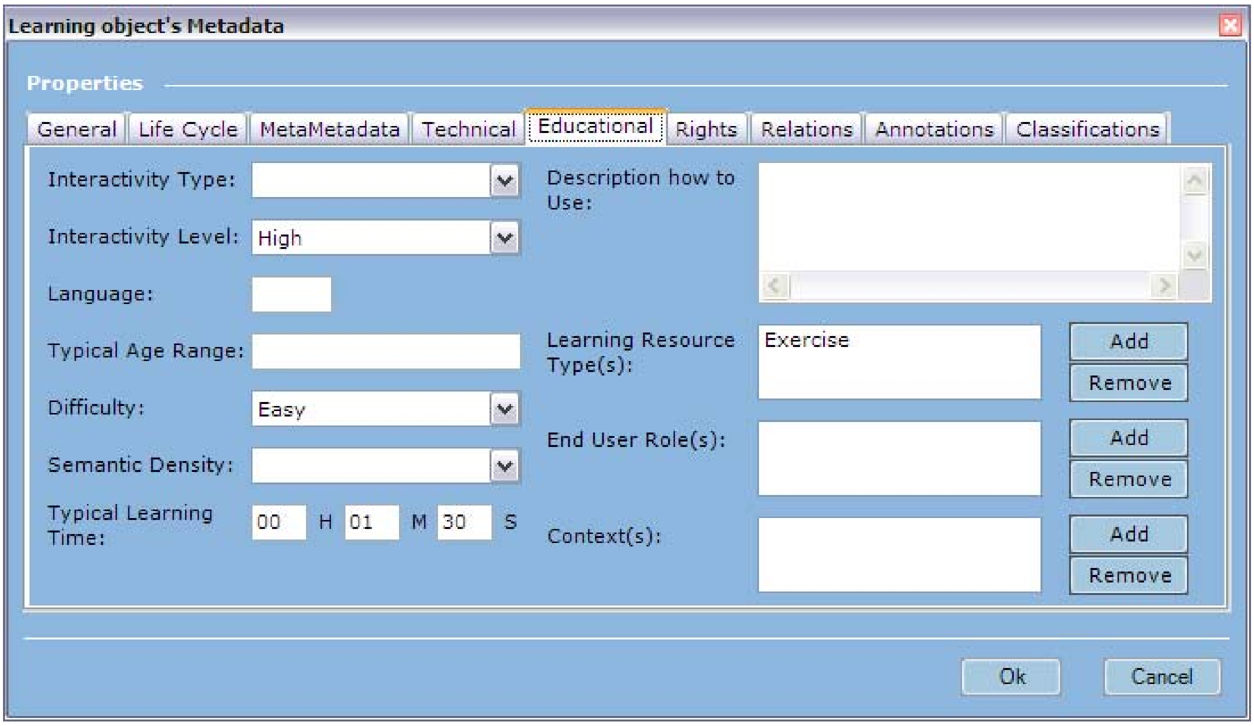


Figure 52. Meta-data entry field for a learning object as provided by the Alfanet authoring tool.

9.5 Preparing questionnaires

9.5.1 KLETT Course:

The following questionnaires are part of KLETT Course:

- Pre-knowledge assessment. An assessment at the beginning of the course in order to locate the learner on appropriate lesson according to his/her knowledge level.
- Lesson assessment. An assessment after every lesson to test whether the learner got the concepts of the lesson and is able to continue with the following lesson.
- Self-Assessment. The user is learning a lesson (activity) and wants to have more practice about it. So, the learner will do the questionnaire on demand (it is optional). The student may be allowed to ask for (self-) assessment any time during his/ her studying of a certain lesson. If s/he does so, the system may show him/her all assessment items related to those LObs, which the student has already marked as done. In this case the system can store the scoring of every item (or set of items), and if the student got the necessary scoring to pass this LOb, this item (or set of items) can be left out in the assessment at the end of the respective lesson.

- Remediation assessment. A questionnaire that is presented to the user in the case s/he has not passes the “lesson assessment”. The Remediation assessment presents to the user in different ways depending on the user profile, i.e. it adapts to the user depending on what modules the user obtained a bad scoring, and the concrete score s/he got.

9.5.2 Defining the QTI adaptation rules

In order to implement questionnaires above mentioned it is necessary the following assumptions:

- to have a bank of items and describe them with associated metadata.
- to define “selection & ordering” rules for the different types of assessments.
- to integrate IMS LD with IMS QTI generating variables in item definition time and in LD definition to manage these variables. Such variables are scoring variables, they could be defined by module, by Learning Objective, by Lesson, etc, depending on the concept that we want to evaluate. In KLETT course all of them was defined, and synchronised between LD and QTI.

9.5.2.1 Metadata definition

The following metadata are included in KLETT items:

- MODULE
  Lessons have defined a set of items for covering a series of modules (Grammar, Reading comprehension, etc…)
- KNOWLEDGE LEVEL (AL)
  Author designs items for several knowledge levels. This is the way for measuring the mastery level of learners. The Knowledge Level of a learner changes according to course execution, so assessments must be adapted to learner knowledge level in that moment. In KLETT course every lesson has a different knowledge level: Lesson 1 = low knowledge level; Lesson 2 = medium knowledge level; Lesson 3 = high knowledge level; Lesson 4 = very high knowledge level
- SUB KNOWLEDGE LEVEL (SUBAL)
  This attribute provides more information about knowledge level of learner. In a remediation scenario, this metadata is useful for generating assessments adapted to “real” learner knowledge level. It intends to express until what extent a learner is getting the concepts of a concrete lesson (i.e. while the knowledge level evolves with the lessons, the sub-knowledge level indicates the knowledge level for a lesson –related with difficulty). Thus, in the Remediation scenario this attribute need to be defined to describe different test items for the same objective but with different difficulty level.
- PURPOSE
  An item could be used for measuring level of mastery of a learner or for practicing an acquired knowledge. Items from one or other purpose could be different.
- ACTIVITY_ID
  It coincides with an identifier of a learning activity. Each item is defined belonging on an activity (or lesson)
- ITEM TYPE
  It could be interesting to have the possibility of generate assessments using specific item types. For instance, for Communicative skills module, could be more suitable to use Free input essay or Fill in the blanks items that other item type.
- LEARNING OBJECTIVE
  Identify the educational objectives covered by such item.
- TEST
  It is a concept very close to Learning Objective.

- LESSON
  KLETT course is composed by Lessons. Each item is defined belonging on a lesson in order to cover the concepts related to such lesson.

The following table summarises all the metadata fields and their possible values used in KLETT course:

Field Name	Field Entry	Description
module	GR	Grammar
	RC	Reading comprehension
	CS	Communicative skills
AL	1	Low level of Knowledge Level (all content of lesson 1)
	2	Medium level of Knowledge Level (all content of lesson 2)
	3	High level of Knowledge Level (all content of lesson 3)
	4	Very high level of Knowledge Level (all content of lesson 4)
SUBAL	Numerical values defined by authors N/A	It is knowledge level at a detailed level . dafault value was defined
purpose	evaluation	Evaluation
	practice	Practice
	survey	Survey
	tutorial	Tutorial
	self-assessment	self-assessment
	final Assessment	Final Assessment
	remediation	Remediation
activity_id	N/A	default value was defined
itemType	True/false	
	Multiple-choice	
	Multiple-response	
	FIB-string	Fill in blank string (with alphanumerical content)
	FIB-numeric	Fill in blank string (with numerical content)
	Image hot-spot	
	Drag-and-drop	
	Essay	
learningObjective	Learning Objective identifier	
test	T1	Reading comprehension for Lesson 1 and Lesson 2
	T2 and T3	Communicative skills for Lesson 1 and Lesson 2
	T4 (T5 and T6)	Grammar in Lesson 1, for Lesson 2 T5 and T6 are valid values too.
lesson	1	For lesson 1
	2	For lesson2

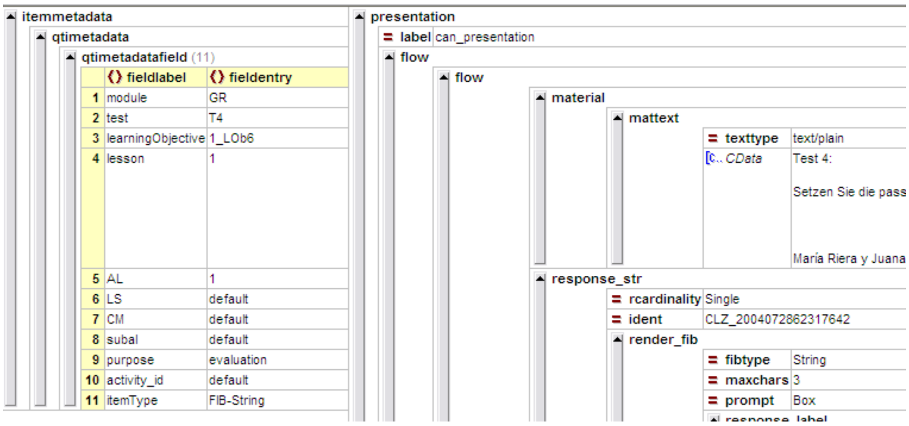


Figure 53. Example of meta-data for a test item.

9.5.2.2 Selection & Ordering rules

Items defined by KLETT are grouped in object banks by lesson. This section describes the Selection and Ordering rules that are defined for the different types of assessments from those object banks.

9.5.2.2.1 Selection & Ordering Rules for generating a pre knowledge assessment:

Selection & ordering rules are:

- 6 items:

  1 from Lesson 1 object bank , with knowledge level (“AL”) 1 and “test” equal to T1

  1 from Lesson 1 object bank, with knowledge level (“AL”) 1 and “test” equal to T3

  1 from Lesson 1 object bank, with knowledge level (“AL”) 1 and “test” equal to T4

  1 from Lesson 2 object bank, with knowledge level (“AL”) 2 and “test” equal to T1

  1 from Lesson 2 object bank, with knowledge level (“AL”) 2 and “test” equal to T3

  1 from Lesson 2 object bank, with knowledge level (“AL”) 2 and “test” equal to T5

- ordered randomly

9.5.2.2.2 Selection & Ordering Rules for generating a lesson assessment:

Selection & ordering rules are:

For Lesson 1

- 4 items:

  1 from Lesson 1 object bank , with “test” equal to T1 but only if test T1 wasn’t passed previously in any self-assessment

  1 from Lesson 1 object bank , with “test” equal to T2 but only if test T2 wasn’t passed previously in any self-assessment

  1 from Lesson 1 object bank , with “test” equal to T3 but only if test T3 wasn’t passed previously in any self-assessment

  1 from Lesson 1 object bank , with “test” equal to T4 but only if test T4 wasn’t passed previously in any self-assessment

- ordered randomly

For Lesson 2

- 6 items:

  1 from Lesson 2 object bank , with “test” equal to T1 but only if test T1 wasn’t passed previously in any self-assessment

  1 from Lesson 2 object bank , with “test” equal to T2 but only if test T2 wasn’t passed previously in any self-assessment

  1 from Lesson 2 object bank , with “test” equal to T3 but only if test T3 wasn’t passed previously in any self-assessment

  1 from Lesson 2 object bank , with “test” equal to T4 but only if test T4 wasn’t passed previously in any self-assessment

  1 from Lesson 2 object bank , with “test” equal to T5 but only if test T5 wasn’t passed previously in any self-assessment

  1 from Lesson 2 object bank , with “test” equal to T6 but only if test T6 wasn’t passed previously in any self-assessment

- ordered randomly

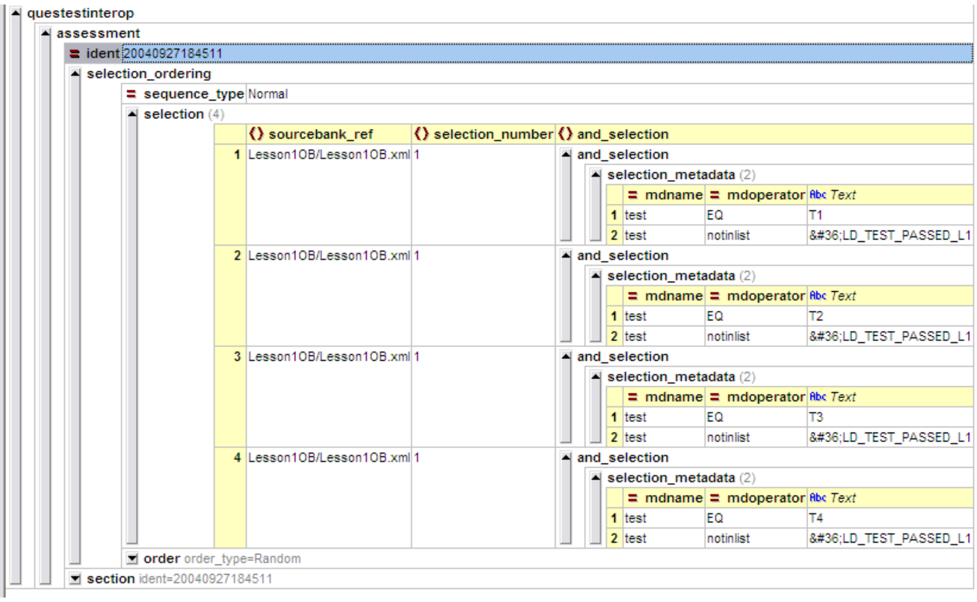


Figure 54. Example of selection and ordering rules for a lesson assessment.

9.5.2.2.3 Selection & Ordering Rules for generating a self-assessment:

Selection & ordering rules are:

- 3 items from object bank
- “test” metadata must be equal to suitable value with according to such case.

For Lesson 1

Module GR: Grammar	T4
Module RC: Reading comprehension	T1
Module CS: Communication skills	T2 , T3

For Lesson 2

Module GR: Grammar	T4, T5, T6
Module RC: Reading comprehension	T1
Module CS: Communicative skills	T2 or T3

- ordered randomly

9.5.2.2.4 Selection & Ordering Rules for generating a remediation assessment:

Every remediation assessments could be defined under the same criteria. It is based in the same selection & ordering rules used in lesson assessments

9.5.2.3 IMS-QTI & IMS-LD Integration

To integrate IMS LD with IMS QTI is necessary to generate variables in item definition time and in LD definition to manage these variables. Such variables are scoring variables, they could be defined by module, by Learning Objective, by Lesson, etc, depending on the concept that we want to evaluate. In KLETT course all of them was defined, and synchronised between LD and QTI and used for influencing the learning process.

For synchronisation purposes a nomenclature was defined for property names. Such nomenclature must be used in IMS-LD design:

sync_[module]_[context]_[count]

Where:

sync: Fixed word to identify what are the properties that should be synchronised

[module]: Identifier of the origin module of the property. In that case is qtiresult

[context]: Contextual identifier to localise the origin of the property (E.g. Context Activity).

[count]: Specific identifier of the property *.

* From each Scoring variable defined in IMS-QTI when the results are synchronised two more variables are created, the maximum scoring and scoring as percentage value. Such variables are named with preffix "max_" and "percen_" respectively. In IMS-LD only percentages was used.

In particular for KLETT course the following scoring variables was defined:

IMS QTI Name	**IMS LD Name**	**Description**
SCORE	sync_qtiresult_[context]_percen_SCORE	General scoring variable
SCORE_L1	sync_qtiresult_[context]_percen_SCORE_L1	Scoring variable for Lesson1
SCORE_L1_T1	sync_qtiresult_[context]_percen_SCORE_L1_T1	Scoring variable for Lesson1 Test1 (Comprehension)
SCORE_L1_T2	sync_qtiresult_[context]_percen_SCORE_L1_T2	Scoring variable for Lesson1 Test2 (Communication)
SCORE_L1_T3	sync_qtiresult_[context]_percen_SCORE_L1_T3	Scoring variable for Lesson1 Test3 (Communication)
SCORE_L1_T4	sync_qtiresult_[context]_percen_SCORE_L1_T4	Scoring variable for Lesson1 Test4 (Grammar)
SCORE_L2	sync_qtiresult_[context]_percen_SCORE_L2	Scoring variable for Lesson1
SCORE_L2_T1	sync_qtiresult_[context]_percen_SCORE_L2_T1	Scoring variable for Lesson2 Test1 (Comprehension)

SCORE_L2_T2	sync_qtiresult_[context]_percen_SCORE_L2_T2	Scoring variable for Lesson2 Test2 (Communication)
SCORE_L2_T3	sync_qtiresult_[context]_percen_SCORE_L2_T3	Scoring variable for Lesson2 Test3 (Communication)
SCORE_L2_T4	sync_qtiresult_[context]_percen_SCORE_L2_T4	Scoring variable for Lesson2 Test4 (Grammar)
SCORE_L2_T5	sync_qtiresult_[context]_percen_SCORE_L2_T5	Scoring variable for Lesson2 Test5 (Grammar)
SCORE_L2_T6	sync_qtiresult_[context]_percen_SCORE_L2_T6	Scoring variable for Lesson2 Test6 (Grammar)

10. Conclusions

Alfanet is the first project that integrates five learning technology standards and as such it is unique. During the development of the system and the course materials for the pilot projects several hurdles had to be taken. One of the most challenging one was the integration and cooperation of IMS-LD, IMS LIP, IMS Metadata and IMS QTI, in particular the integration and cooperation of IMS LD with IMS-QTI. Although both specifications come from the same organisation little was known beforehand how such an integration should be realised. But once the exchange of data between the two modules was a fact, it provided multiple possibilities to further refine the adaptive features of the Alfanet system. As a result the current version of the Alfanet system includes a variety of adaptation scenarios based on a tight integration and cooperation of the selected standards. The scenarios can be used directly or as a reference to others involved in systems or standards design and show the feasibility of a generic standards-based framework for e-learning. Additionally, the Alfanet system and/or its components (available as Open Source) are at a stage that they can used for further development and exploration of adaptive-standards based e-learning.

Designing courses using multiple learning technology specifications without an integrated toolset was difficult to manage and time consuming for the authors. The authoring tools for each individual component helped the authors, which were most of the time not experts on the learning technology specifications used, to create their course parts. Nevertheless, it is clear that for real-world usage the authoring process has to be further simplified and more tightly integrated unless there are dedicated experts available to support part of the authoring process.

11. References

EML, (2000). *Educational modelling language 1.0*, retrieved, February 21, 2005 from http://hdl.handle.net/1820/81

Greeno, J.G., Collins, A.M., & Resnick, L.B. (1996). Cognition and Learning. In D.C. Berliner, & R.C. Calfee (Eds.), Handbook of Educational Psychology (pp. 15-46). New York: Simon & Schuster Macmillan

Hermans, H., Manderveld, J., & Vogten, H. (2004). Educational Modelling Language. In W. Jochems, J. van Merrienboer, & R. Koper, *Integrated e-Learning*. London: Routledge Falmer, 80-99

Hernández L, D., Asensio Pérez, J.I., Dimitriadis, Y. (2004), IMS Learning Design Support for the Formalization of Collaborative Learning Patterns. Proceedings of the the 4th International Conference on International Conference on Advanced Learning Technologies, Joensuu, Finland.

IEEE LTSC, (2000). *Standard for Information Technology --Education and Training Systems -- Learning Objects and Metadata.* Retrieved February 21, 2005 from http://ltsc.ieee.org/wg12/index.html

IMSLD (2003). IMS Learning Design. Information Model, Best Practice and Implementation Guide, Binding document, Schemas. [available online at: http://imsglobal.org].

IMSMD (2003). IMS Meta Data. Information model. [available online at: http://imsglobal.org]

IMSQTI (2004). IMS Question and Test Interoperability. [available online at: http://imsglobal.org]

IMSLIP (2003). IMS Learner Information Profile. [available online at: http://imsglobal.org]

IMSCP (2004). IMS Content Packaging. [available online at: http://imsglobal.org]

Koper, E, J, R. & van Es, R. (2004). Modeling Units of Learning from a Pedagogical Perspective In McGreal (Eds.) *Online Education Using Learning Objects (Open and Flexible Learning).* Canada: RoutledgeFalmer

Koper, E, J, R. (2001). *Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML,* retrieved, February 21, 2005 from http://hdl.handle.net/1820/36

Koper, E, J, R., & Manderveld, J. M. (2004). Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology*, 35 (5), 537-552.

Koper, R., & Olivier, B. (2004). Representing the Learning Design of Units of Learning. *Educational Technology & Society*, 7 (3), 97-111

Littlejohn, A. (Eds.). (2003), Reusing Online Resources: A Sustainable Approach to eLearning. London: Kogan Page.

Pedagogical Perspective In McGreal (Eds.) *Online Education Using Learning Objects (Open and Flexible Learning).* Canada: RoutledgeFalmer

Rawlings, A., Rosmalen van, P., Koper, R., Rodríguez-Artacho, M., Lefrere, P., (2002). *CEN/ISSS WS/LT Learning Technologies Workshop: Survey of Educational Modelling Languages (EMLs).* Retrieved January18, 2005 from http://sensei.lsi.uned.es/palo/eml-version1.pdf

Reigeluth (Ed.), Instructional-Design Theories and Models: a New Paradigm of Instructional Theory, Volume II (pp. 5-29). Mahwah: Lawrence Erlbaum Associates, Publishers.

Reigeluth, C.M. (1999). What is Instructional-Design Theory and How is it Changing? In C.M.

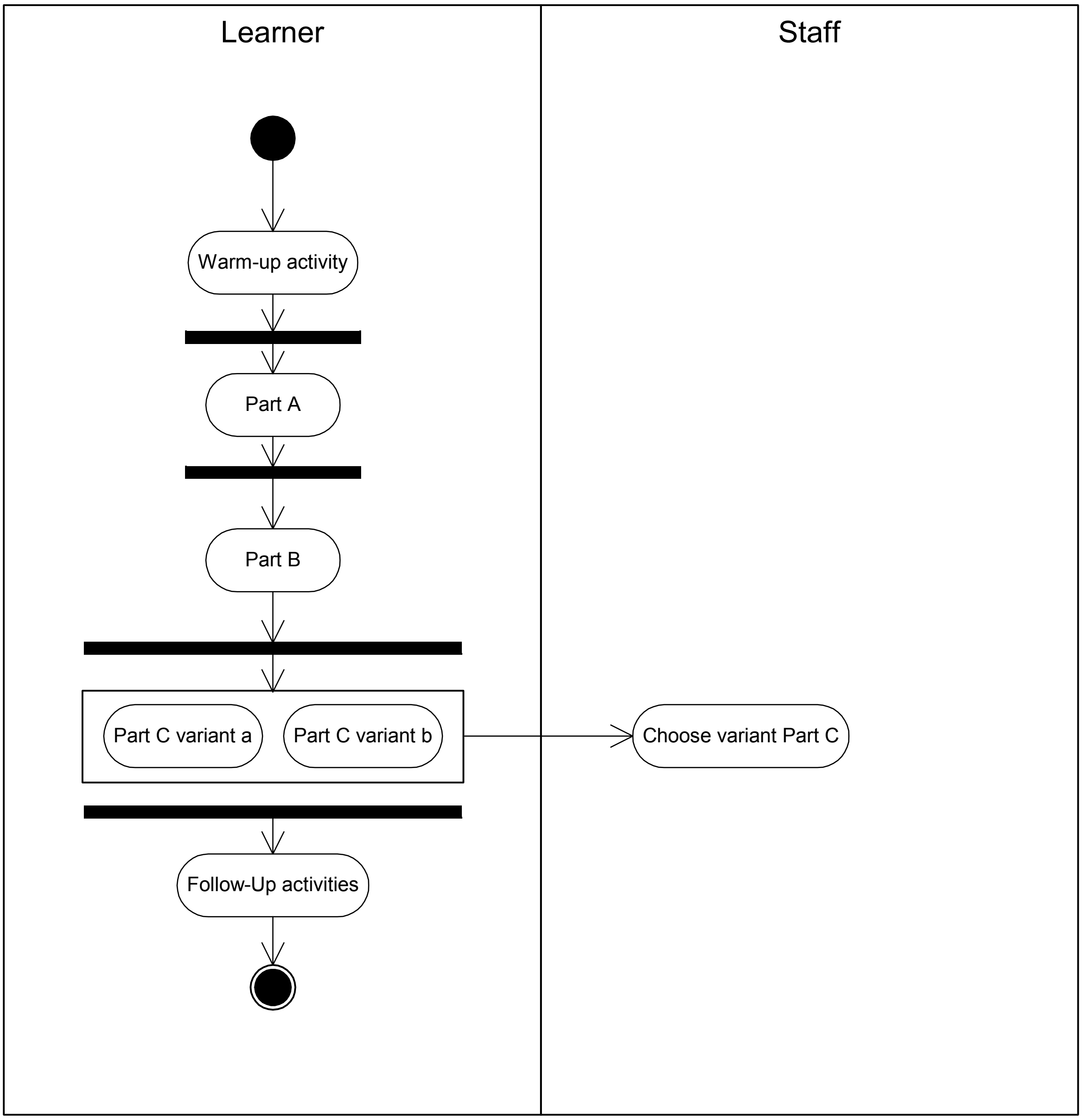
Reigeluth, Ch.M. (ed.). (1983). Instructional design theories and models: an overview of their current status. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stolovitch, H.D., & Keeps, E.J. (Eds.) (1999). Handbook of Human Performance Technology. San Francisco: Jossey-Bass Publishers.

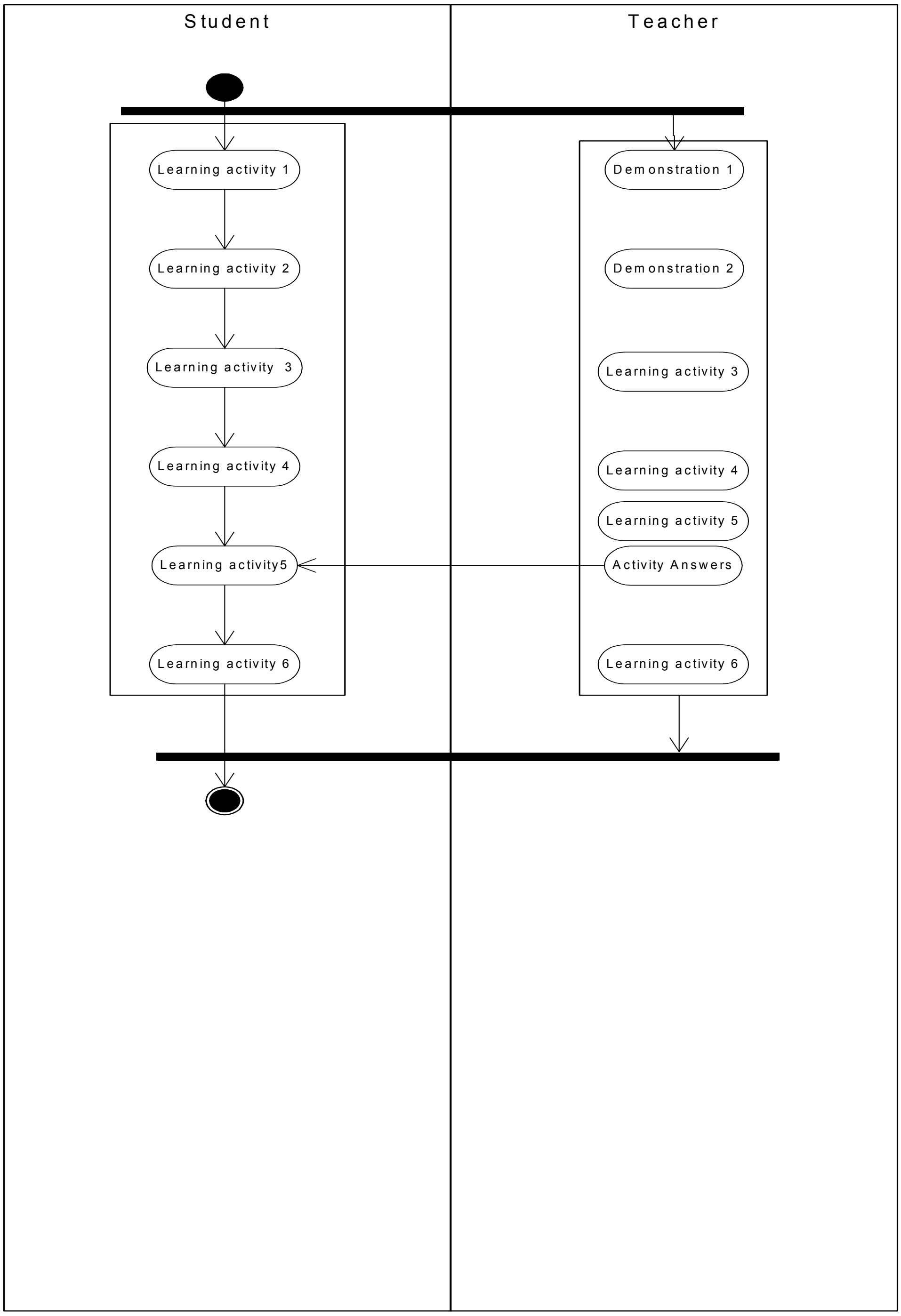
Weber, R. P. (1985). Basic content analysis. Beverly Hills: Sage.

Appendix 1: Activity diagrams of the used lesson plans

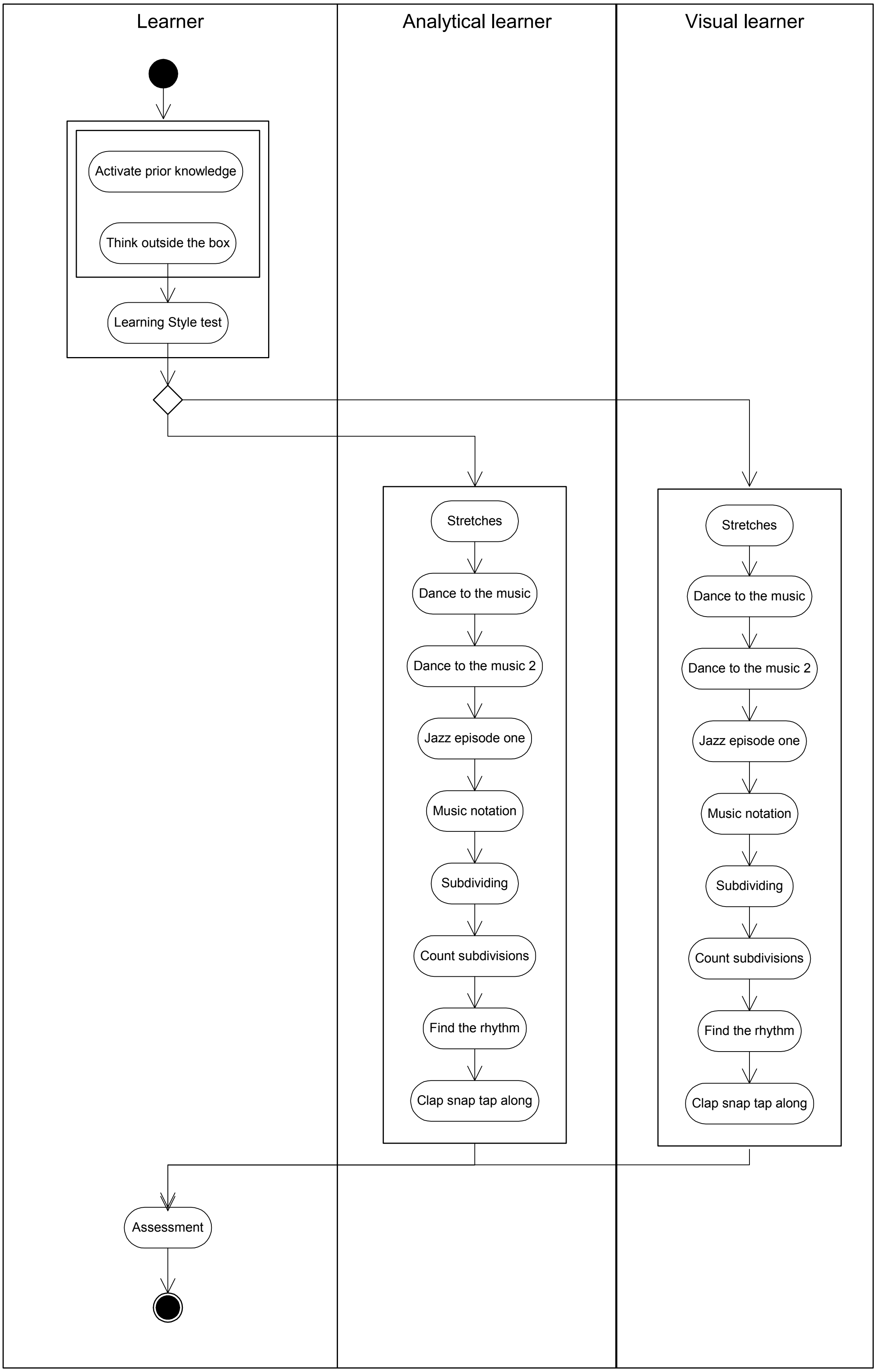
Lesson plan 1



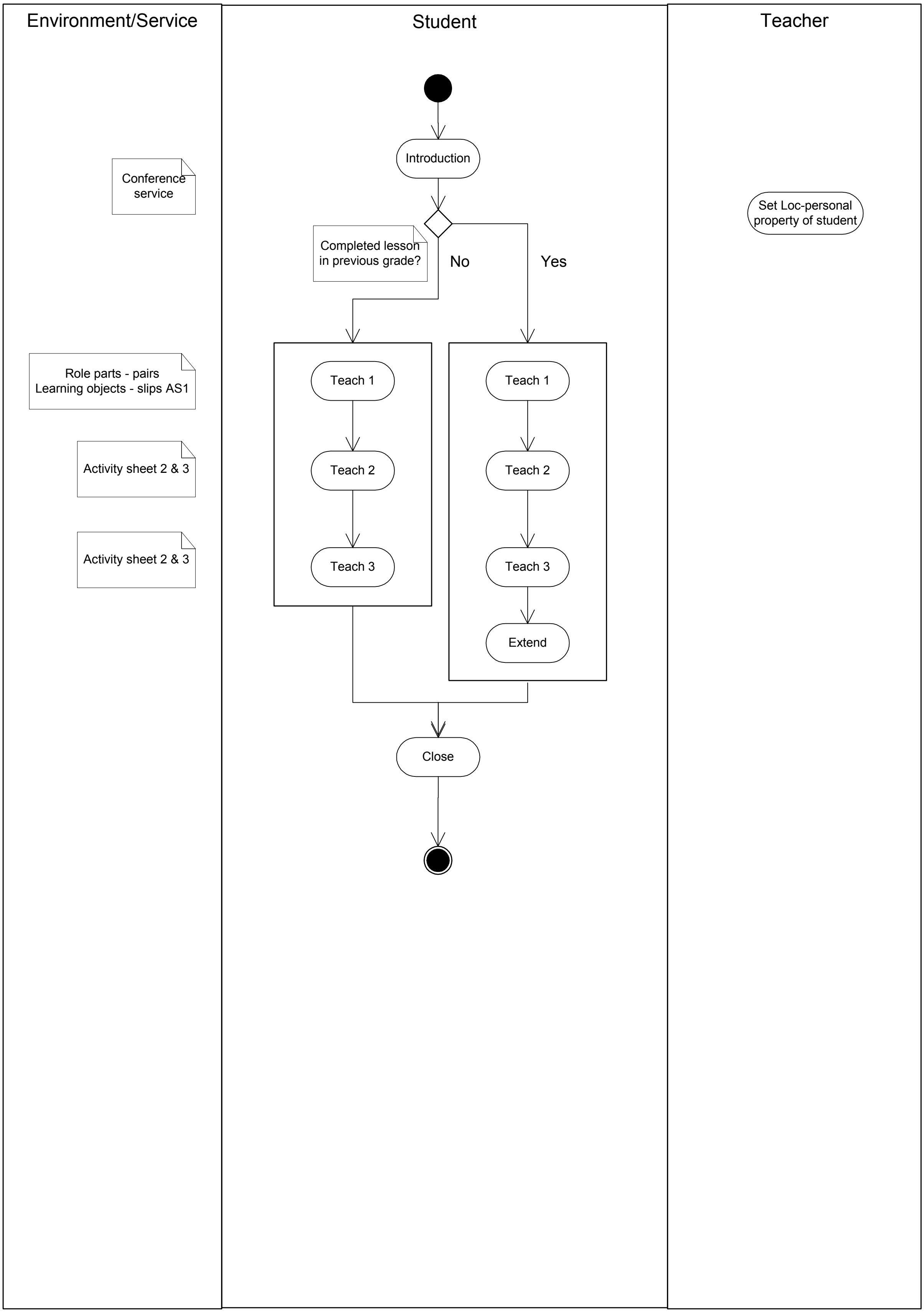
Lesson plan 2



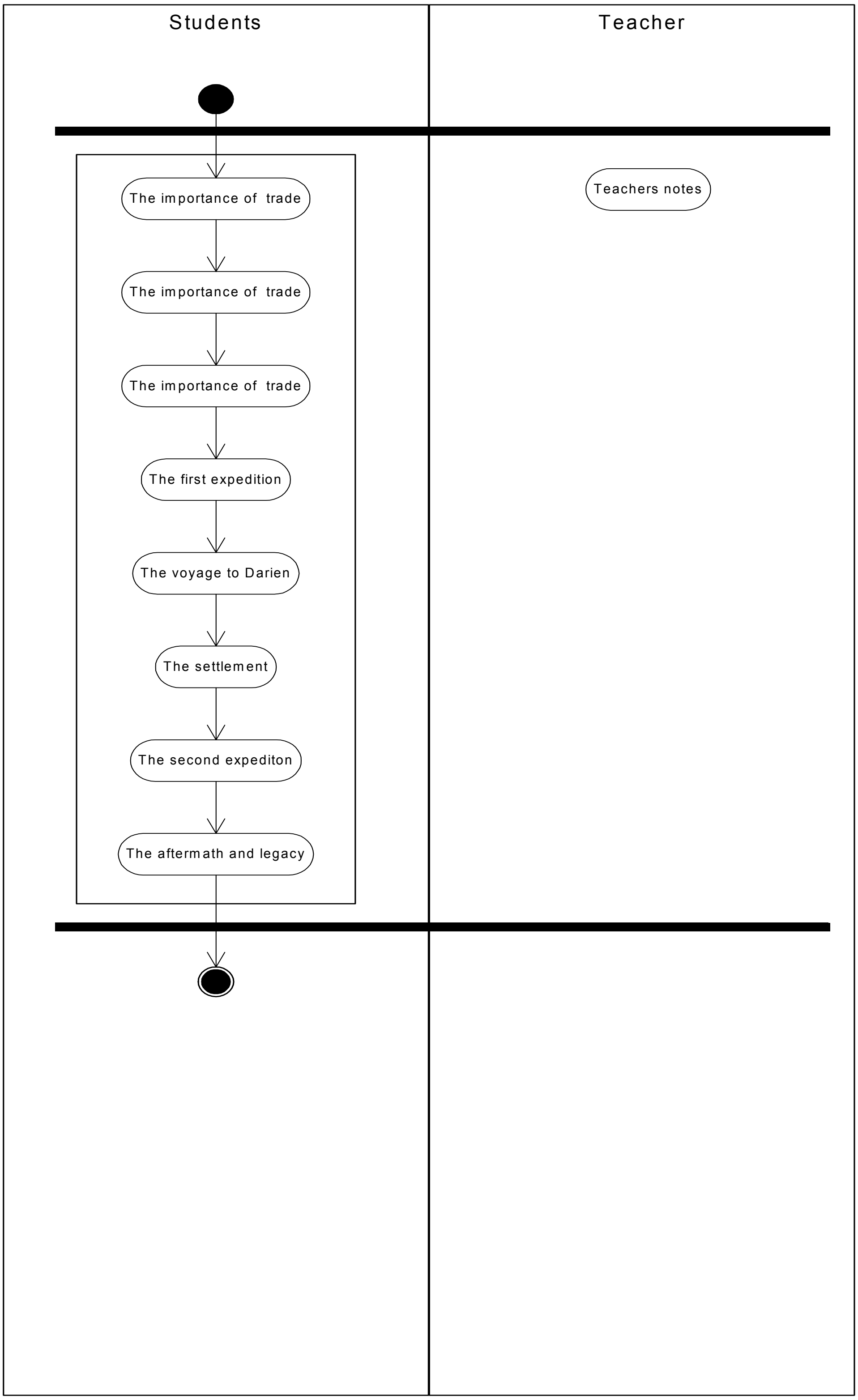
Lesson plan 3



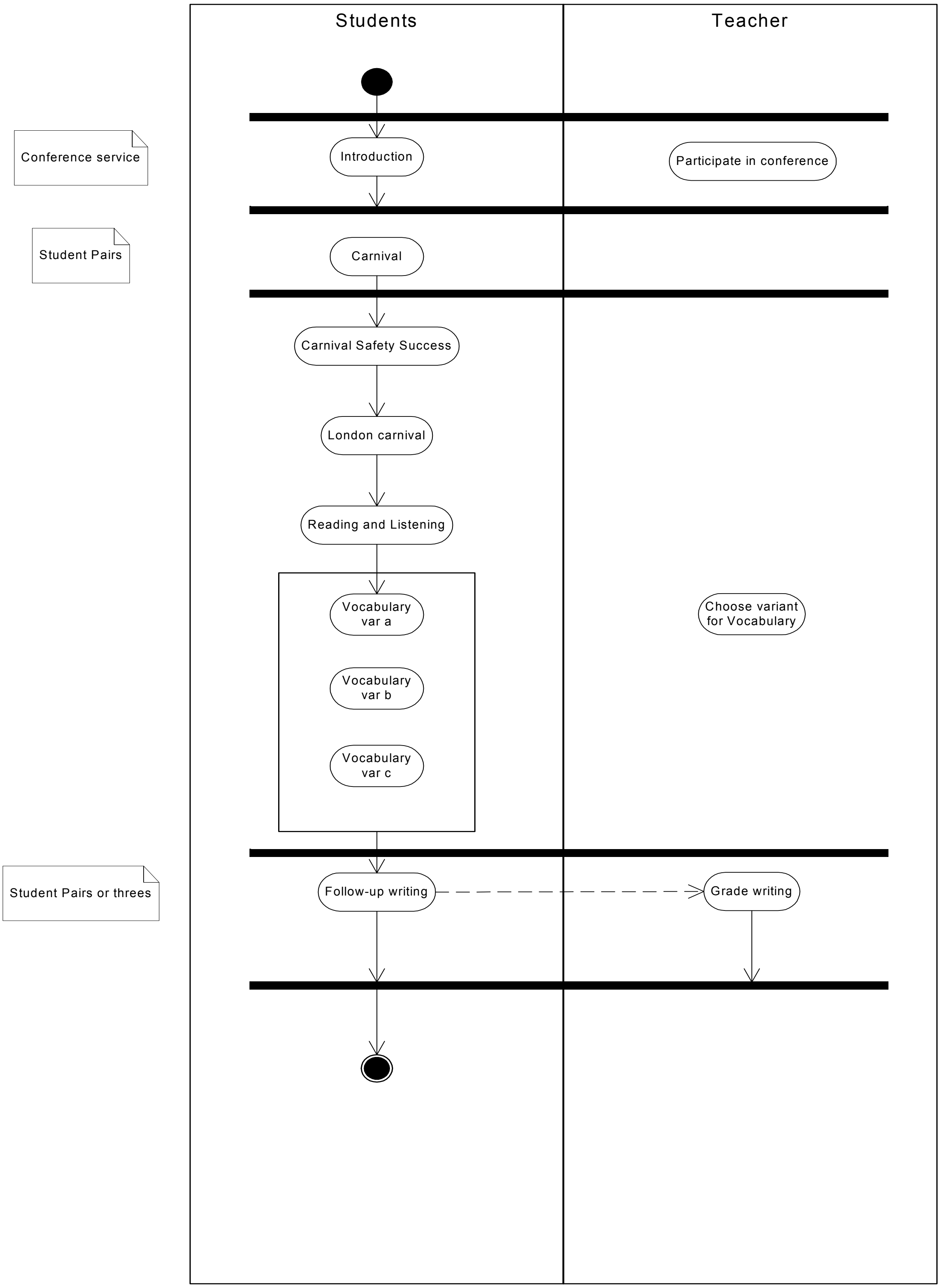
Lesson plan 4


Environment/Service
Student
Teacher
Introduction
Conference service
Set Loc-personal property of student
Completed lesson in previous grade?
No
Yes
Role parts - pairs Learning objects - slips AS1
Teach 1
Teach 2
Teach 3
Teach 1
Teach 2
Teach 3
Extend
Activity sheet 2 & 3
Activity sheet 2 & 3
Close

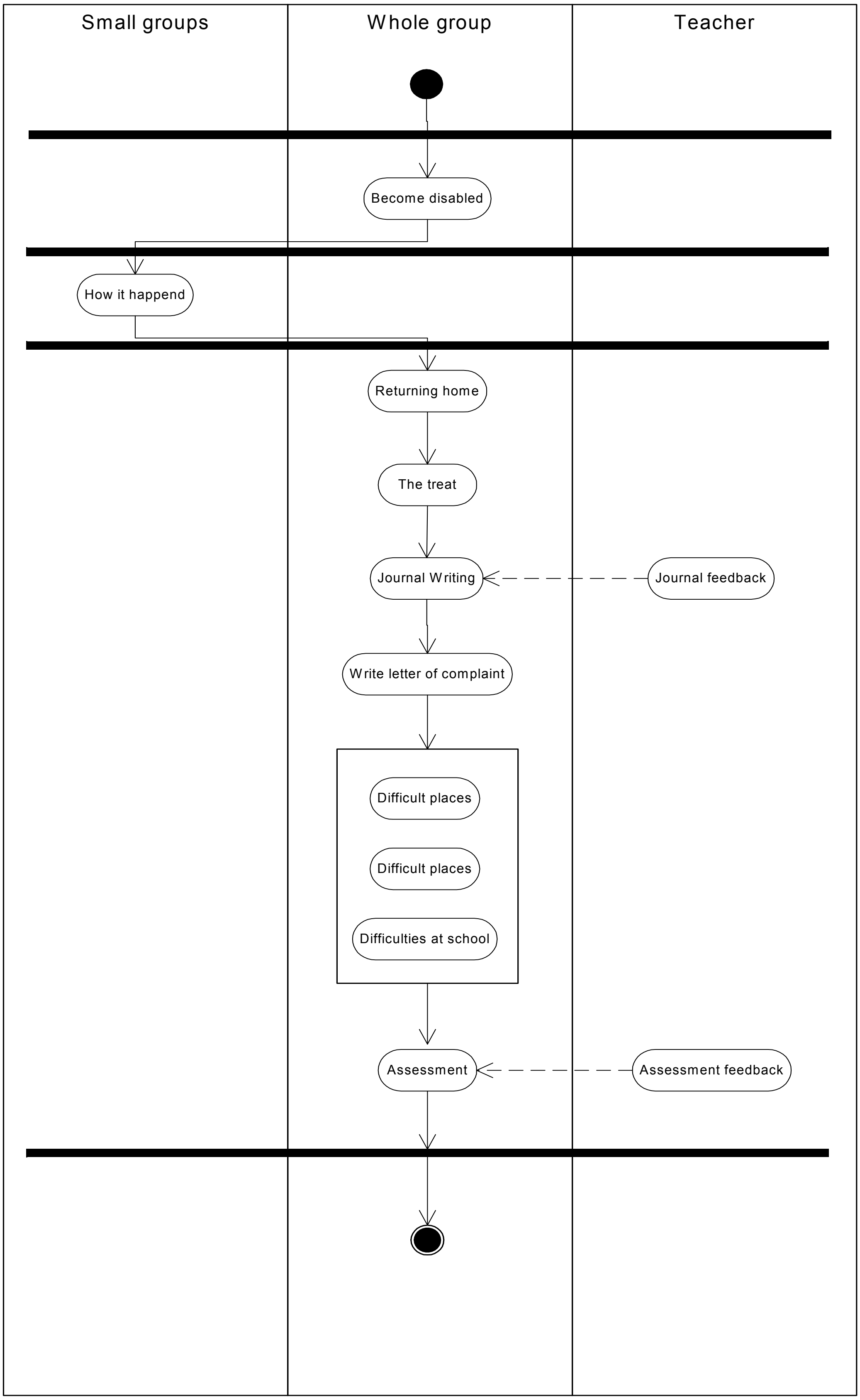
Lesson plan 5



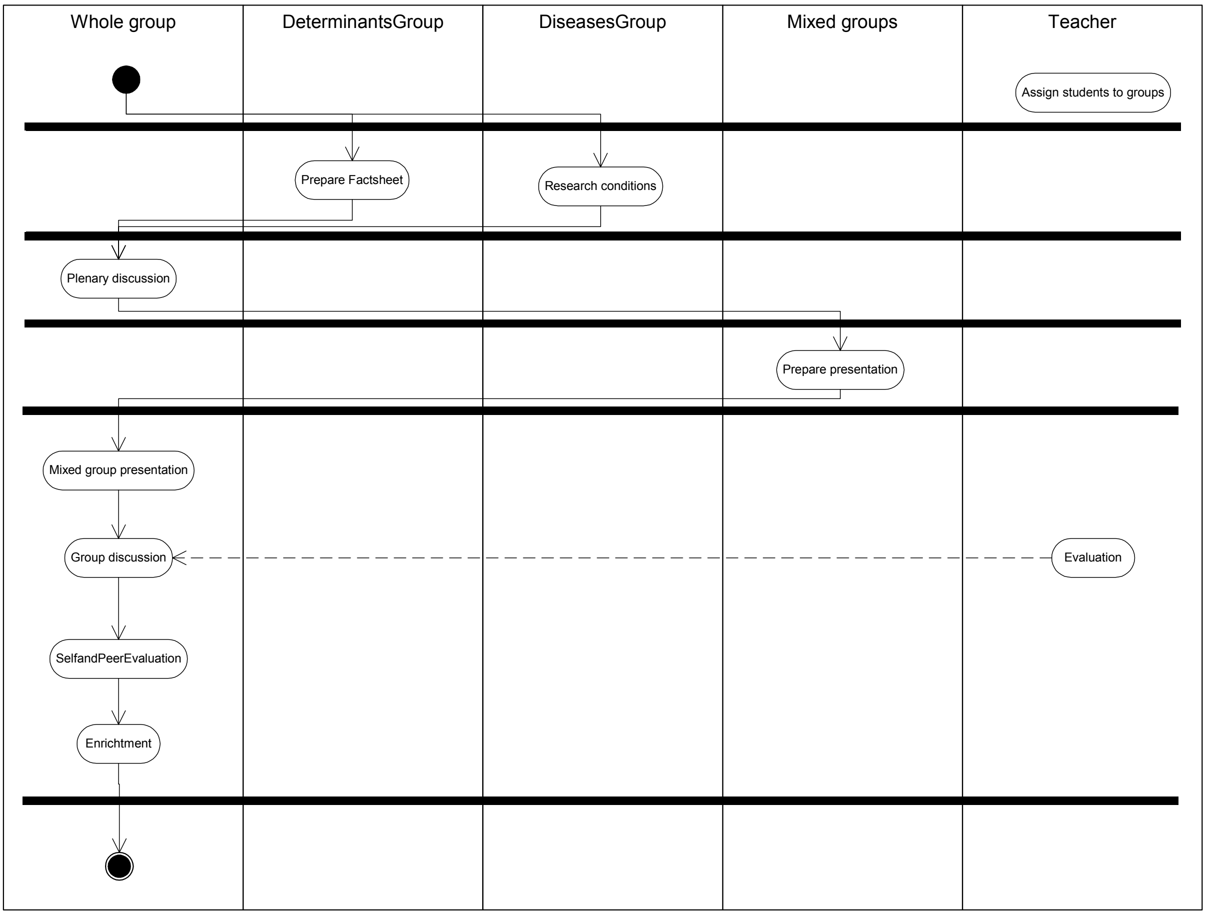
Lesson plan 6



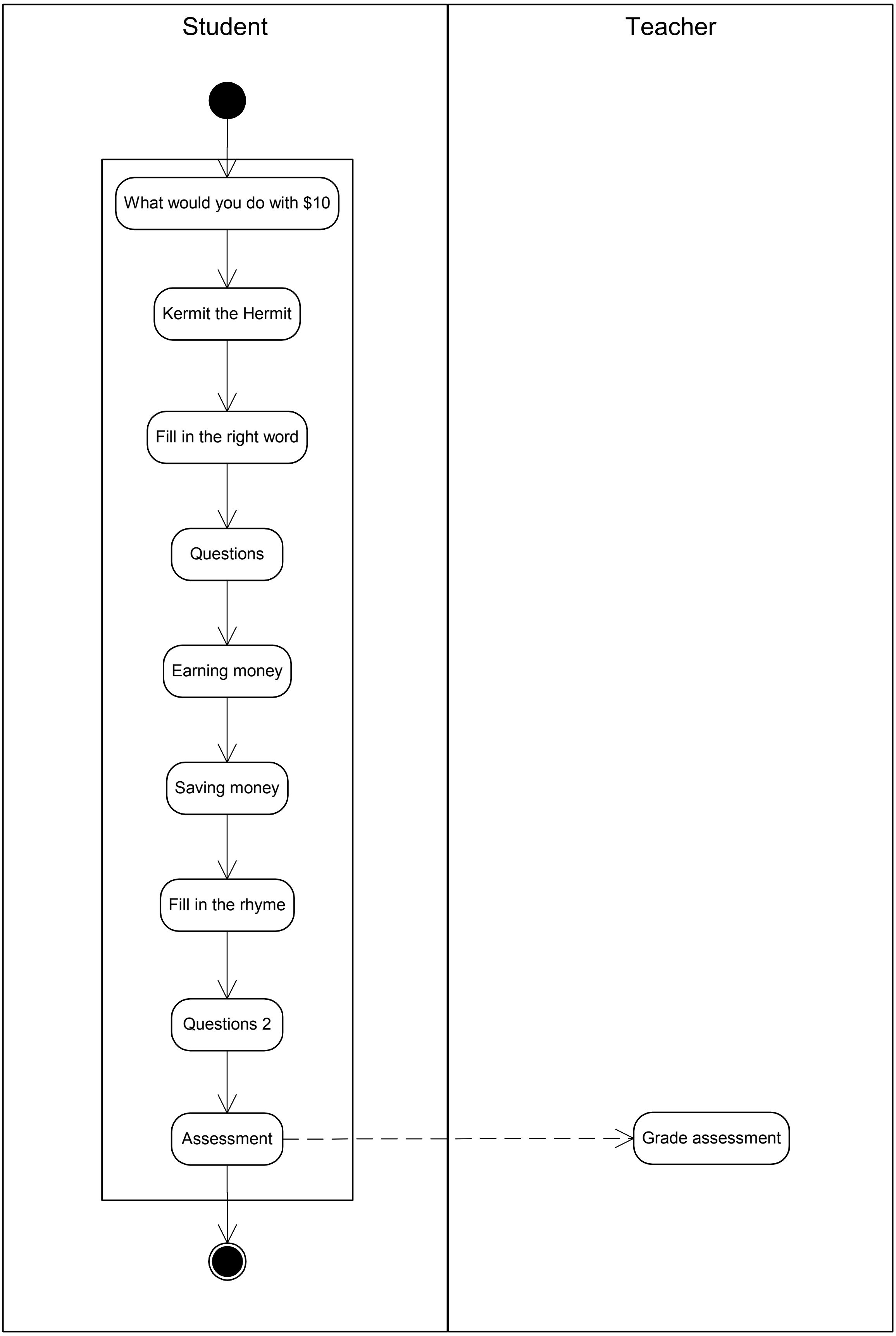
Lesson plan 7



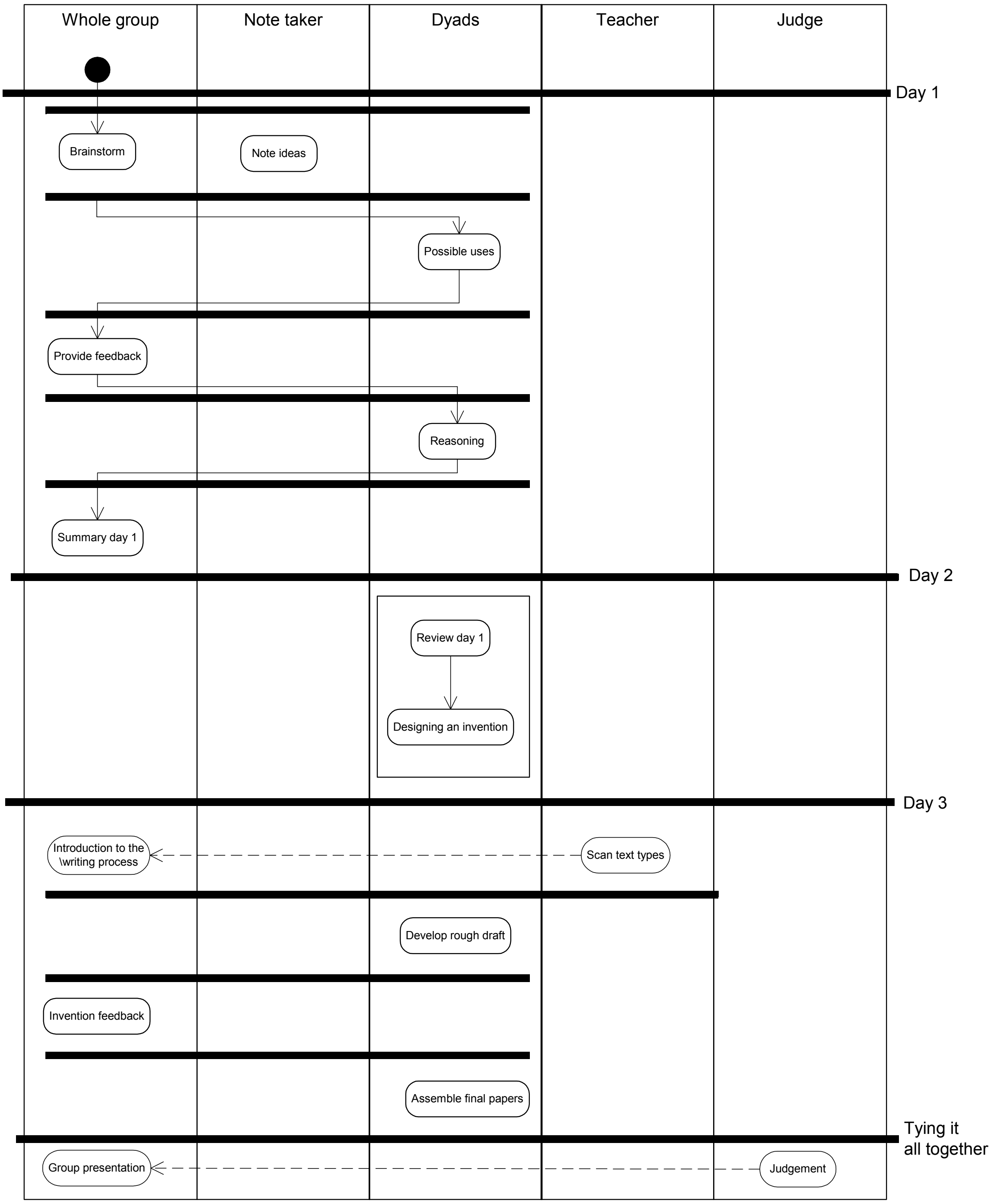
Lesson plan 8



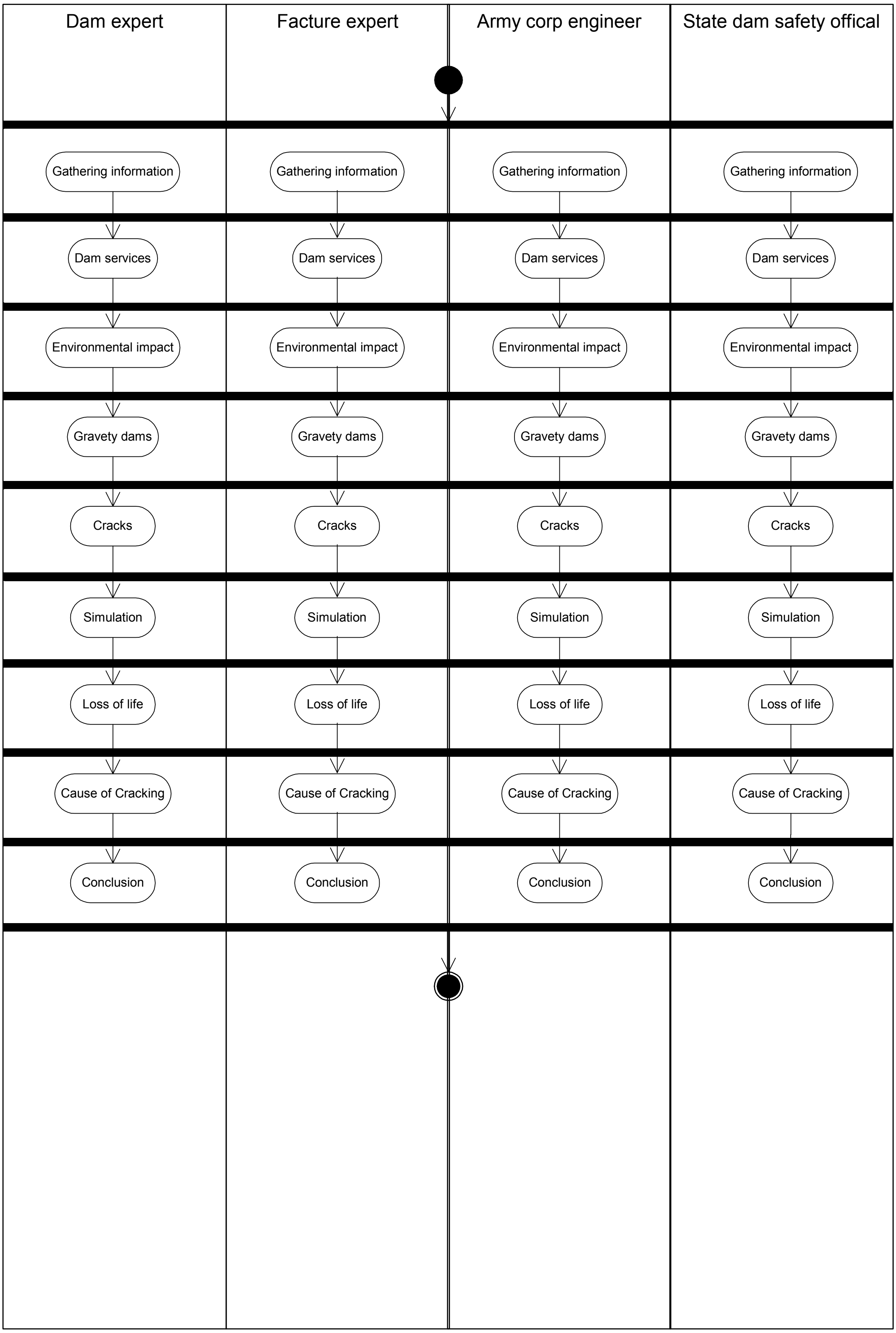
Lesson plan 9



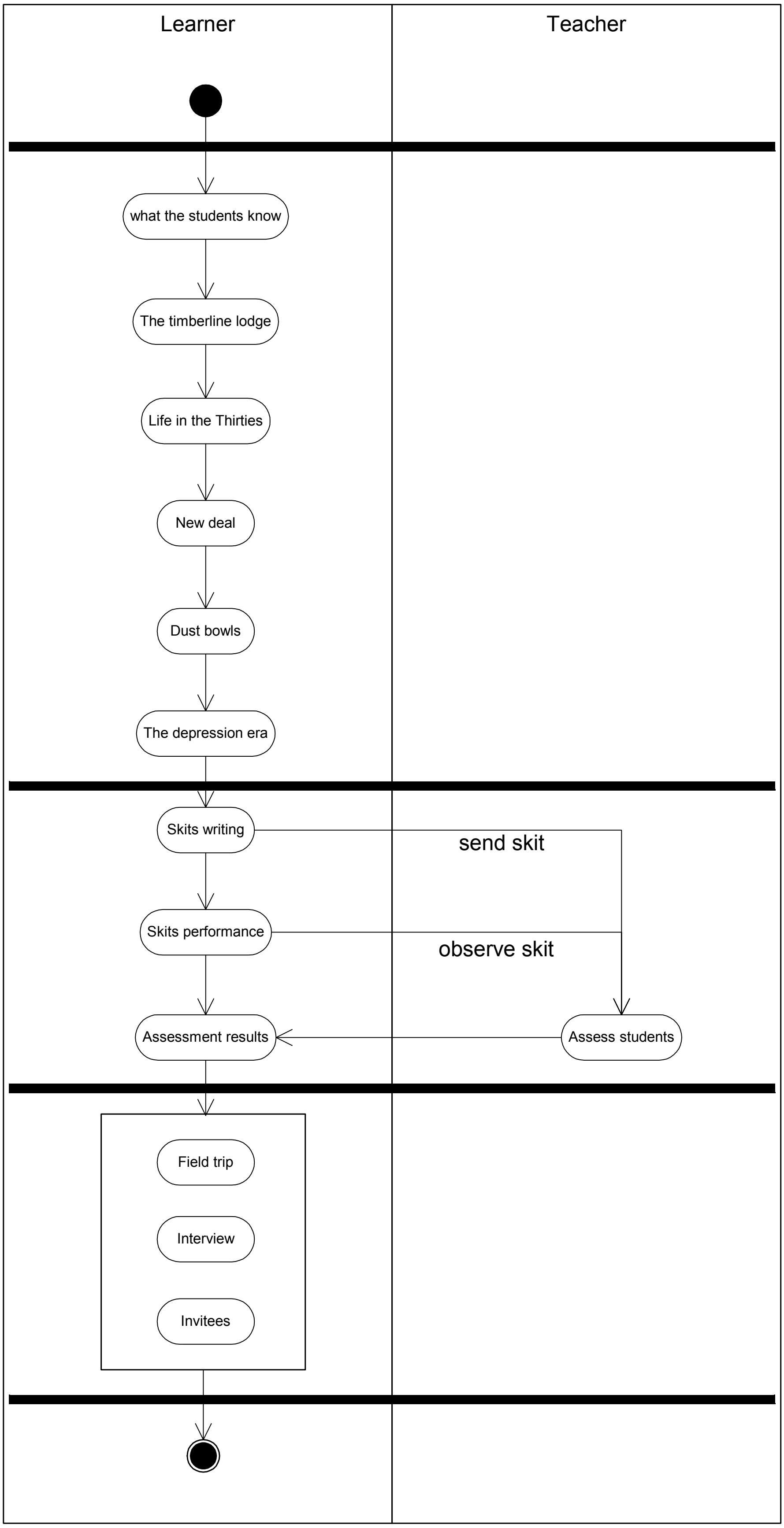
Lesson plan 10



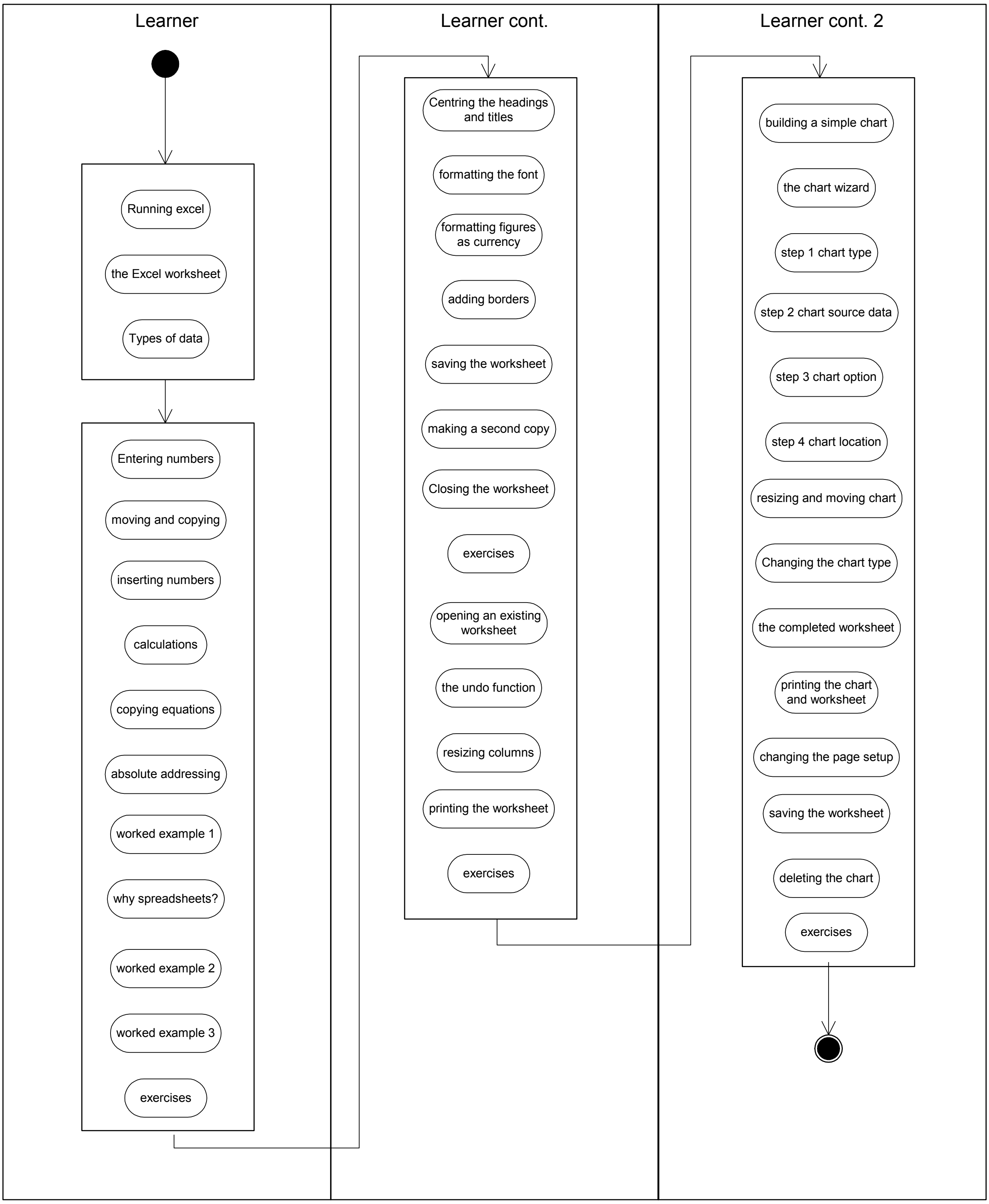
Lesson plan 11



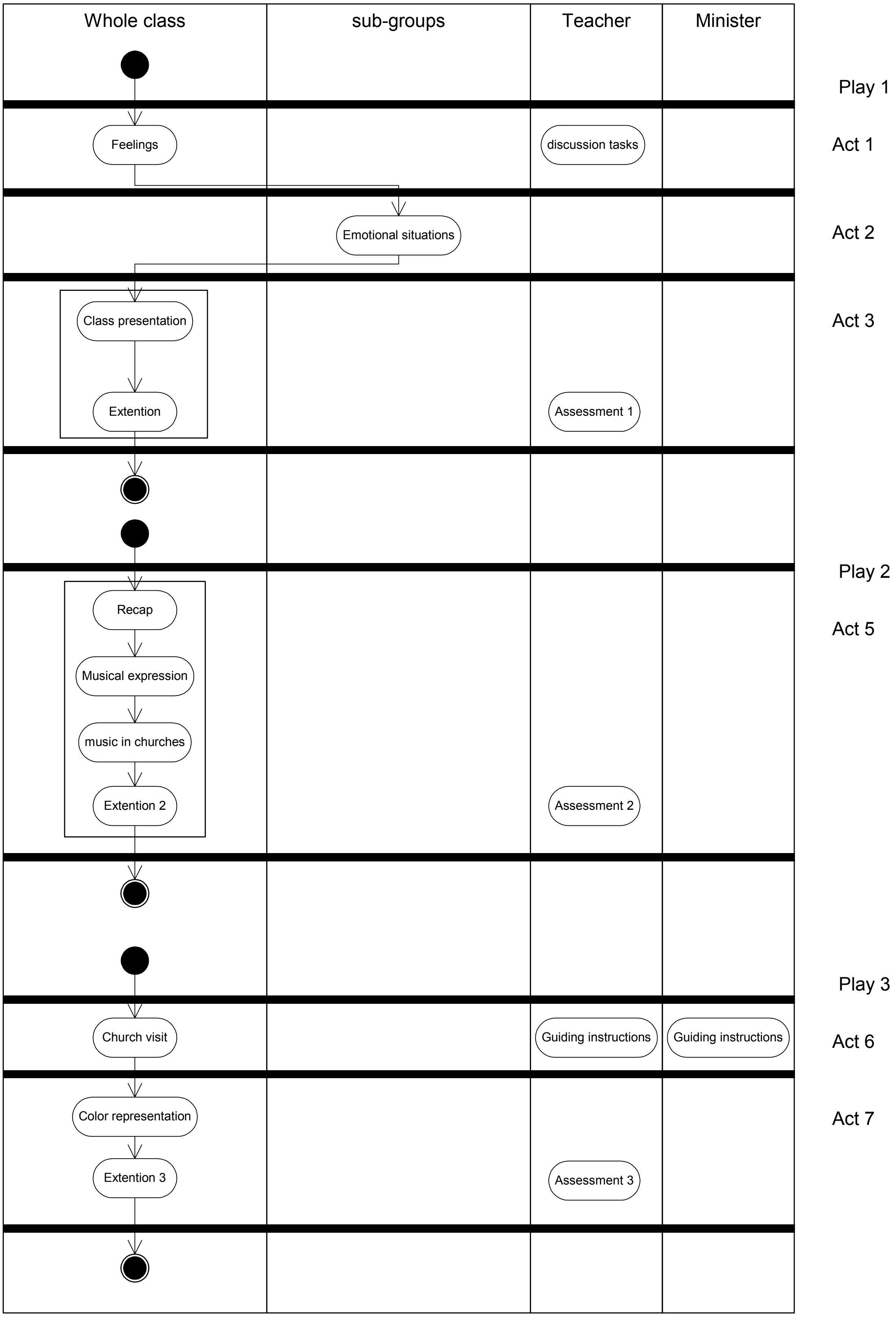
Lesson plan 12



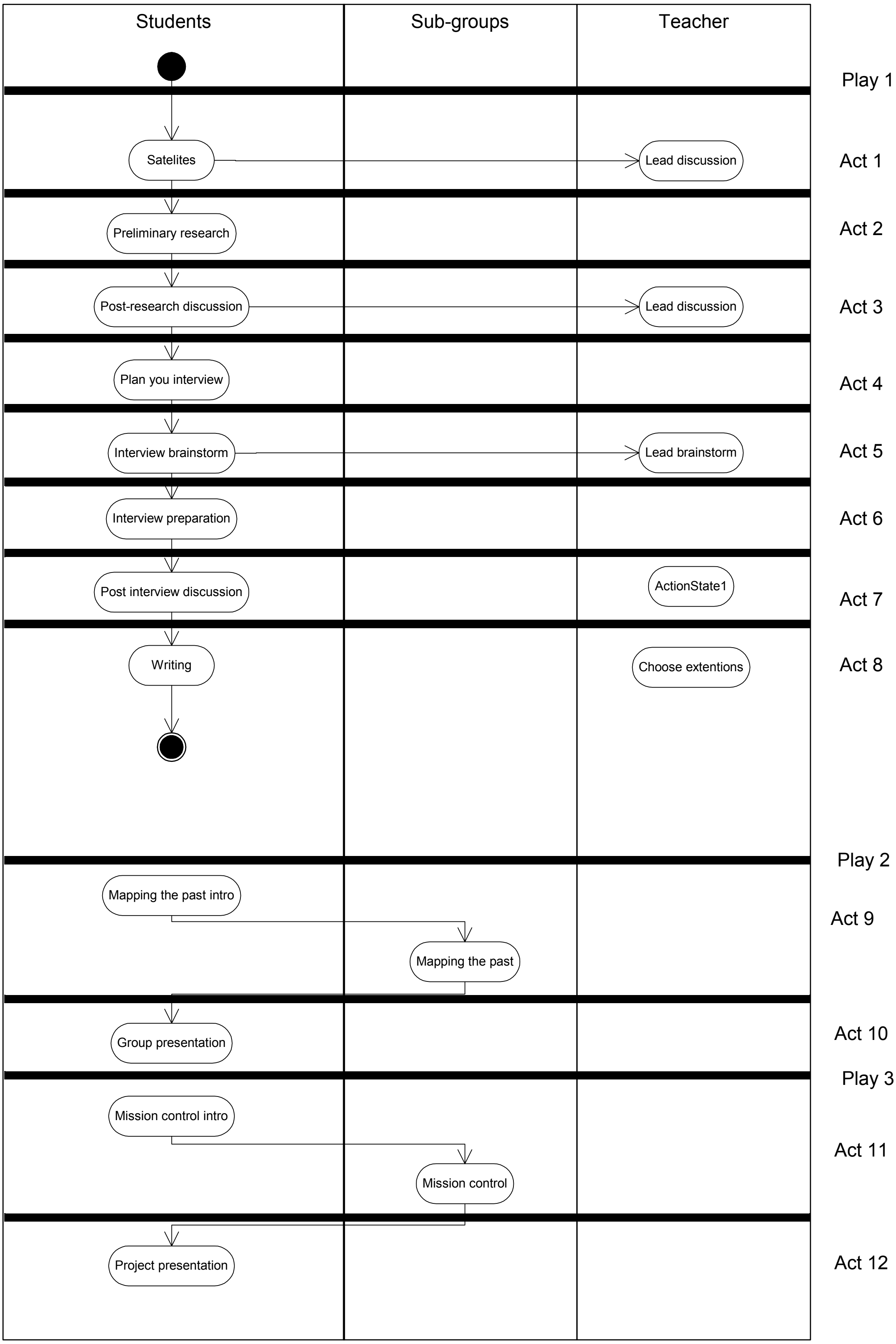
Lesson plan 13



Lesson plan 14



Lesson plan 15



Lesson plan 16