How to use an Instructional Engineering Method and a Modelling Tool

Gilbert Paquette, Ileana de La Teja, Michel Leonard, Karin Lundgren-Cayrol, Olga Marino

Chapter 6

Using an Instructional Engineering Method and a Modeling Tool to Design IMS-LD Units of Learning

Gilbert Paquette, Ileana de la Teja, Michel Léonard, Karin Lundgren-Cayrol, Olga Marino

CIRTA(LICEF) Research Centre, Télé-université, Montréal

ABSTRACT

This chapter discusses how to build IMS learning designs focusing on three aspects, instructional engineering, modeling tools and graphical design techniques. First, we propose that instructional designers use a systemic and systematic instructional engineering method to build Units of Learning conforming to the IMS-LD specification. MISA, a mature instructional engineering method will serve as the basis to our design approach. Second, we present a graphical modeling tool, MOT+, and a representation technique that was created to support instructional engineering. In MOT+, concepts, procedures and principles are used to describe all IMS-LD components as well as their relationships. We believe this graphical language to be closer to instructional designers, in that it represents a more pedagogical viewpoint than software engineering graphical languages like UML, while still enabling an automatic translation from graphical models into a machine-readable IMS-LD XML. Third, we will provide an example of the design processes involved in building learning designs, from the preliminary analysis to the definition of a unit of learning method, the central part of the IMS Learning Design.

Introduction

The fast evolution of learning technologies has multiplied the num-

ber of decisions one must take to create a distributed learning system (DLS). While it is true that a majority of the first Web-based applications have been mostly used to distribute information, more and more educators have become aware of the need to go beyond simple uses of information and communication technologies. This context has generated a much-needed interest for pedagogical methods and, more generally, for the field of Instructional Design (Wiley 2002).

The term "Educational Modeling Language (EML)" was first introduced in 1998 by researchers at the Open University of the Netherlands (OUNL), as a response to Instructional Design and pedagogical concerns towards standardization and interoperability needs. The work on Educational Modeling Languages (Koper 2001), and the subsequent integration of a subset in the IMS Learning Design Specification (IMS 2003a), is the most important initiative to date, to integrate Instructional Design preoccupations into the international standards movement. In particular, it describes a formal way to represent the structure of a Unit of Learning and the concept of a pedagogical method specifying roles and activities that learners and support persons can play using learning objects.

The IMS-LD specification leaves open the choice of instructional methods and modeling tools that can support designers in the process of building learning design specification, especially for those aiming at distributed, networked or on-line education. Extensive research and development in the field of Instructional Design has led to a large body of methodologies. We believe that the Instructional Engineering approach (Paquette 2001a) and the Learning Systems Engineering Method (MISA[1]) is especially well suited to help designers build IMS-LD compliant Units of Learning.

This chapter is structured into four sections. Section 1 presents the instructional engineering viewpoint on the IMS-LD specification. Section 2 outlines the MISA instructional engineering method and its relation to IMS-LD. Section 3 presents the MOT+ graphical rep-

[1] MISA is the French acronym for Méthode d'ingénierie des systèmes d'apprentissage

resentation language and situates MISA/MOT+ as embedding an educational modeling language with its XML machine-readable output. Section 4 presents a practical learning design case of a complex unit of learning.

1. Instructional Engineering viewpoint on the IMS-LD specification

Instructional Engineering can be defined as "*A method that supports the analysis, the design and the delivery planning of a learning system, integrating concepts, processes and principles of instructional design, software engineering and knowledge engineering"* (Paquette 2003, p. 56).

1.1 Defining Instructional Engineering

Located at the crossroads of instructional design, software engineering and knowledge engineering, from which it inherits most of its properties, Instructional Engineering, is a particular systemic and systematic method in the field of educational problem solving. It is founded on the system sciences (Le Moigne 1995; Simon 1973) that defines the concept of a system as a series of units in dynamic interaction, organized in order to achieve specific goals.

The origin of *instructional design*[2] goes back to John Dewey (1900), who, a century ago, claimed the development of an "interlinked science" between learning theories and educational practices. Since the fifties, the evolution of this new discipline has been carried by influential researchers such as B.F. Skinner (1959), Jerome Bruner (1966) and David Ausubel (1968). In the seventies and eighties, *instructional theories* have blossomed through the work of researchers such as Gagné (1970), Scandura (1973), Merrill(1976), Landa

[2] In American literature, this discipline is known as "Instructional Design (ID)", "Instructional System Design (ISD)" or "Instructional Science" (Reigeluth, 1983; Merrill, 1994) depending on theoretical inclination. In Europe, one of the pioneers of the field used the term "Scientific Pedagogy" (Montessori, 1958).

(1976), Reigeluth and Rogers (1980), Collins and Stevens (1983), to name a few. These instructional design models and theories have been built on solid foundations and present an impressive body of work. However, today it seems necessary to renew the instructional design methods and tools to support the creation of Distributed Learning Systems (DLS) that are heavily dependent on information and communication technologies.

Software engineering brings some interesting solutions to meet demands required by innovative technology used in DLS. From a technical point of view, a Unit of Learning, and its distributed environment, is an information system consisting of a complex array of software tools, digital documents and communication services. This environment allows learners and facilitators to interact using information and communication technologies. By adapting software engineering principles to instructional design principles, Instructional Engineering proposes well-defined processes and principles that help produce deliverables, precisely described products of these processes. Moreover, multi-agent systems offer a good way to represent the enacted learning designs at delivery time as a set of agents, persons and digital objects, interacting to help some of the agents to learn and others to facilitate learning.

Knowledge engineering is a methodology developed in the field of expert systems and artificial intelligence over the last thirty years. Knowledge engineering focus on identifying and structuring knowledge to explain it, using a symbolic or graphical language representation to facilitate its use by persons and/or computer systems. Knowledge engineering has been applied in education to build intelligent tutoring systems [Wenger, 1987] and also as support systems for designers [Merrill, 1994; Spector et al., 1993]. Recently, the focus has shifted to machine-readable knowledge structures aiming at a new generation of the Web (Berners-Lee et al, 2000). In an *Instructional engineering* method, knowledge modeling processes or the workflow are at the forefront. The workflow model guides the designer in his tasks to define content and objectives using them as an orientation for the design of instructional scenarios, learning ob-

jects (or educational resources)[3], as well as the learning system delivery processes.

1.2 Relationship between Instructional Engineering and the IMS Learning Design specification

Developing high quality distance learning courses can be a difficult and expensive task. On-line course development faces two main challenges: viability and quality. A key concept has emerged as a response to the concern of viability, the concept of reusability. Basically, reusability means being able to use an educational resource or learning object (LO) in different educational contexts or courses, possibly supported by different independent or interoperated e-learning delivery systems, which demands for a standard way of describing those learning objects. In the past few years, a vast movement towards international standards for learning objects has been initiated. Duval & Robson (2001) present a review of the evolution of standards and specifications starting with the Dublin Core metadata initiative in 1995 up to the publication of the Learning Object Metadata (LOM) standard in 2002. A host of other specifications have been published since then.

But what about quality? High quality learning objects are necessary but not sufficient to produce a high quality course or unit of learning. When, how, for what and by whom will those LOs be used? The IMS-LD specification offers a standardized way to associate learning materials (learning objects), activities and actors in a learning scenario. Furthermore, having an XML format that can be read by any compliant delivery system, IMS-LD bridges the gap between the process of designing a course and that of delivering it. What is still needed, to ensure quality of a course, is to ensure the quality of the learning scenarios produced by the design process. Basically, instructional engineering methods like MISA, and tools like MOT+ and ADISA[4] guide and support course designer(s) through the proc-

[3] We will use here the terms learning object, educational resource or simply resource as synonyms throughout this chapter.

[4] ADISA (Distributed Workshop for Learning Systems Engineering) is tools developed at Télé-université. It is a web-based system that supports course de-

ess of designing high quality learning systems and scenarios, in particular, by ensuring coherence through systematic documentation of all aspects of the design process and products, automatic propagation of many pieces of information as well as a systemic view of the process.

Figure 1 presents a general view of the relationship between instructional engineering methods and tools, and EML/IMS-LD specifications. The remaining part of this chapter focuses on a presentation of MISA as an instructional engineering method and MOT+ as a modeling tool to support this process. In Chapter 16, we discuss the DLS delivery process by analyzing Explor@, an open system for learning and content delivery developed at the Télé-université in Quebec.

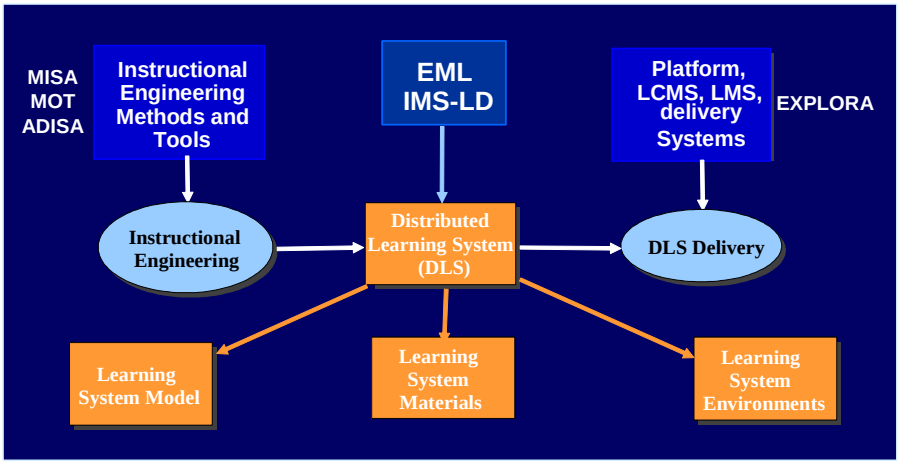

Figure 1– Interrelations between MISA 4.0, IMS-LD Design and Explor@

2. An Instructional Engineering Method for Learning Design

signing teams in the elaboration and integration of the various elements of the MISA method.

Implementation

This section presents a synthesis of our work in Instructional Engineering at Télé-université in Québec (Canada). We will present the main MISA 4.0 Instructional Engineering Method components and concepts, and then introduce a more detailed description of the design processes inherent to the instructional model, which in turn will assist instructional designers in producing IMS-LD compliant Units of Learning.

2.1 The MISA 4.0 Instructional Engineering Method

A knowledge modeling approach is used to define the Instructional Engineering method itself, its concepts, processes and principles. This R&D initiative started in 1992 and has led to the MISA 4.0 version (Paquette 2001a, 2002a) and to its support tool, called ADISA (Paquette et al 2001). The editor MOT+ is embedded in the ADISA system and accessible through a web browser from any workstation linked to the Internet.

MISA is based on a problem solving approach. The Method starts by (1) identifying the educational problem, its context and constraints as well as general orientations, (2) defining preliminary solution, (3) building the LS architecture including elaboration of the knowledge and competency model as well as the instructional model, (4) designing instructional materials, (5) modeling, producing and validating learning materials and (6) specifying LS delivery model(s) as well as maintenance and quality management. The 6 phases in MISA are illustrated in Figure 2.

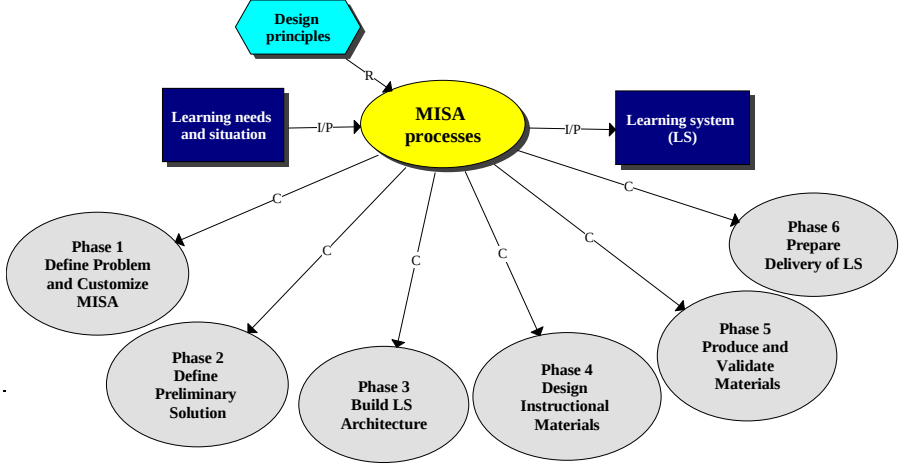

Figure 2: The Main MISA Process and its 6 Phases

The whole process is guided by a set of design principles that must be taken into account when building high quality distance learning systems:

- *Self-Management and Meta-cognition principles:* Explicit association of a skill to a set of knowledge units, where the skill's generic process guides the design. Offer different learning paths and personalization options to be self-managed by learners. Promote self-management by introducing support tools like progress reports. Provide explicit meta-cognitive activities, such as for example individual and group product and process formative task evaluation.
- *Information processing principles***:** Include rich and diversified static and dynamic information resources, clearly related to activities. Provide access to search, annotation, and modeling tools to manipulate resources as well as production tools adapted to each task.
- *Collaboration principles:* Collaborative and individual activities must sustain one another. Adapt the modalities of collaboration to the generic process in which the collaboration is proposed. Allow for both synchronous and asynchronous interactions. Provide management tools for coordinating collaborative activities within the LS.
- *Personalized Assistance principles*: Encourage heuristic and methodological guidance rather than algorithmic assistance. Including multiple facilitators, both human and machine, to provide a flexible learning environment. Provide assistance mainly on the learner's initiative.

In each of the phases 2 to 6, MISA also proposes the development along four axes: knowledge, instructional, learning materials and delivery model.

The *Knowledge Model* centers on a graphical representation of the

Learning System's content domain. In this model, the domain's facts, concepts, procedures and principles are displayed and interrelated with precise links. Then target and prerequisite competencies are associated to units of knowledge, thus identifying prerequisites and learning objectives for the Instructional Model. Subsequently, knowledge units and competencies are attributed to learning units, instruments or resources used in the learning units.

The *Instructional Model* is essentially a network of Learning events and units, to which knowledge and target competencies are associated. Each learning unit is described by a learning scenario specifying learning and support activities linked to resources in the environment. Resources holding content (called instruments) are associated with a subset of the knowledge model.

The *Learning Material Models* are useful to describe materials (learning objects), their media components, source documents and presentation principles as well as other specifications aimed at graphical designers and learning material producers.

Finally, *Delivery Models* are produced to show how and where actors use or provide the learning materials and resources such as tools, communication means, services and locations, used in the learning system. Each Delivery Model is a multi-user workflow, where actors use or produce resources, while assuming different roles. These processes correspond to organizational issues, such as group organization, staff assignments, technical help, resource delivery, and so on, which must be prepared to ensure smooth network-based or distance learning deployment.

The MISA Learning Engineering process produces specifications grouped in documentation called Design Elements (DE), resulting from sub-tasks in the 6 phases presented in figure 2. These DE are also organized according to the four axes within each phase. Presently, MISA 4.0 comprises 35 basic sub-tasks, each producing one DE, numbered, as shown in table 1, from 100 to 640. The first digit denotes the phase, the second, the axis or model, and the third, the sequence number within the axis.

The first task in each axis (shown in table 1) aims to define orientation principles pertinent to the axis model and based on the general principles stated in the Problem Definition phase. These principles help define one or more graphical models (***bold italics*** in table 1) built using the MOT+ knowledge representation technique and tool (Paquette 1999, 2002b). Graphical models are the basic DE in each axis, the backbone of the MISA method. Most of the other tasks, in MISA, describe properties of objects in these models (e.g., competencies, learning units, resources, roles) as well as their relationships. MISA also includes revision and validation tasks in Phase 5, which allow the cyclic evolution of the learning system design and reduce the risk of costly errors. Phase 6 mainly serves to specify the deployment and delivery aspects of the learning system.

Problem Definition		
100 Organization's Training System 102 Training Objectives	104 Target Populations 106 Actual Situation	108 Reference Documents
Knowledge Model	**Instructional Model**	
210 Knowledge Model Orientation Principles ***212 Knowledge Model*** 214 Target Competencies ***310 Learning Unit Content*** ***410 Learning Instrument Content*** 610 Knowledge/Competency Management	220 Instructional Principles ***222 Learning Event Network*** 224 Learning Unit Properties ***320 Instructional Scenarios*** 322 Learning Activity Properties 420 Learning Instrument Properties 620 Actors and Group Management	
Learning Materials Model	**Delivery Model**	
230 Media Principles 330 Development Infrastructure 430 Learning Materials List ***432 Learning Material Models*** 434 Media Elements 436 Source Documents 630 Learning System/Resource Management	240 Delivery Principles 242 Cost-Benefit Analysis 340 Delivery Planning ***440 Delivery Models*** 442 Actors and User's Materials 444 Tools and Telecommunication 446 Services and Delivery Locations 540 Assessment Planning 542 Revision Decisions Log 640 Maintenance/Quality Management	

Table 1 – MISA 4.0 Design Elements/Tasks and Products by Axes

2.2 MISA Instructional Model

An Instructional Engineering method like MISA involves the interaction of many specialists such as content experts, instructional designers, media producers and training managers (see also Chapter 8). Each of these main actors is central to one of the four axes, but they all interact and intervene in all axes as well. We will now focus on the instructional model axis, where the instructional designer is the main actor.

In producing design element 220, the instructional designer will set a number of orientation principles, formulate a learning metaphor, identify the type of learning event network or course structure, specify types of learning scenarios, collaboration, content assessment (see chapter 7), resources, documents, services, the degree to which activities can be customized and any other instructional principles, which could help construct the global learning design corresponding to the educational problem. Seventeen typologies have been thoroughly researched and integrated in the MISA method's support documentation as well as in ADISA.

Based on these principles, the instructional designer will proceed to design element 222, where he will construct the learning design's instructional model, called the *Learning Event Network*, which is a generic term to describe a module, a course, a training program, etc. In IMS-LD, it corresponds to the structure of the Method, that is, information on number of Plays, Acts and Activity-structures included in the Unit of Learning.

In MISA, a Learning Event Network is composed of learning events (LE) and/or learning units (LU) (which are terminal learning events), resources, links and rules. Composition links (C) are used to represent the hierarchy of nested learning events, also seen as the course structure. The precedence (P) link is used to indicate whether a LE/LU is prerequisites to another. Resources are inputs (link I/P going in) to Learning Events/Units or their products (link I/P going

out). Rules express the conditions applied (link R) to Learning Events/Units, for instance, a choice to be made between alternative Learning Events/Units or a specification of the kind of evaluation, collaboration or adaptation that will take place during the Learning Event/Unit. Figure 3 shows an example structure of the Course: Equipment Maintenance, which is composed of five modules, where four are terminal learning events and thus called Learning Units, and one is a Learning Event, decomposed into two Learning Units.

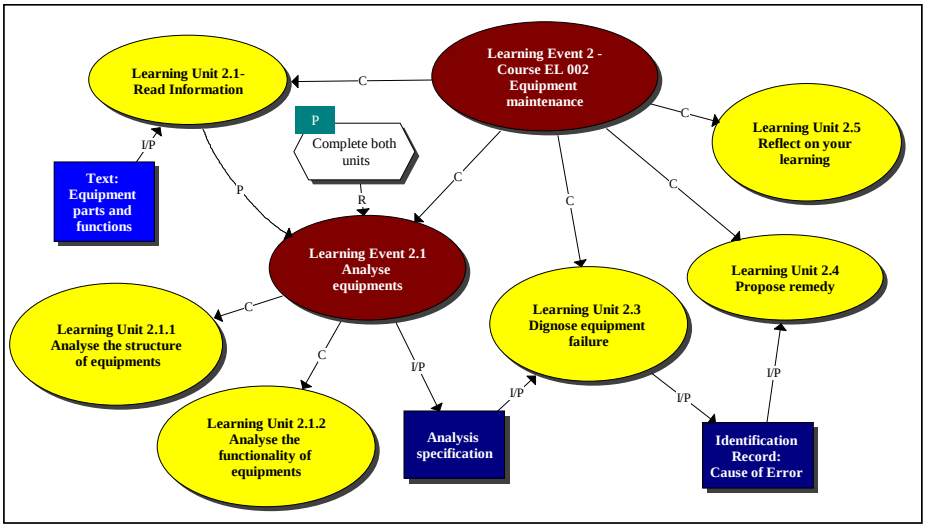

Figure 3 Example of a MISA Learning Event Network

Each Learning Unit consists of one Instructional Scenario describing the relationship among actors (facilitators and learners), activities and resources,. The set of activities performed by learners is called the Learning Scenario. It includes all required and produced resources, links and rules. The set of activities performed by facilitators (ex.: tutors, teachers, evaluators, etc). is called the Assistance Scenario.

The next step is to build a *learning scenario model* for each Learning Unit, where the designer takes into account target and entry as

well as prerequisite competencies, which were all defined in the Knowledge Model. We have shown elsewhere (Paquette 2001a) that it is possible to derive the learning scenario from a generic skill proposed in the target competency (or in a learning objective) for that learning unit. For example, if a target competency states that learners should learn to diagnose equipment failures, a generic diagnostic process will provide a workflow or task model composed of the individual diagnostic tasks including their inputs, products, and control principles.[5]

An Assistance scenario is created when the designer adds to this basic flow of tasks, an instructional intervention strategy. For example, in an expository approach, an instructor will use the workflow model to present segments of the diagnostic process. In a constructivist approach, diagnostic problems concerning equipment failure will be proposed to the learners and the instructor will use the diagnostic workflow model to give advice to learners carrying out the tasks.

MOT+ graphical models use ovals to represent procedures. In instructional scenario models, they are used to represent activities that are performed by actor roles that are represented by small hexagons holding the letter L for learner or F for facilitator (equivalent to staff in IMS-LD). Rectangles represent resources in the environment, labeled I for instruments, T for tools, S for services, and C for communication means. Unmarked resources are outcomes produced by the actor during an activity. White hexagons represent the four kinds of rules labeled P for progression, E for evaluation, C for collaboration and A for adaptation rules. R-links are used to relate actors to activities. For resources an I/P-link is used, ingoing/outgoing to/from an activity. Activities can be linked to other activities by precedence links (P-link) expressing a sequence of activities. Rules found in the Learning Event Network model are also used in the Instructional Scenario model. Rules of progression, evaluation, collaboration and adaptation are represented by a hexagon and can be R-linked to activities.

[5] This approach is similar to the KADS software engineering methodology (Breuker et al, 1999)

Figure 4 illustrates a MISA instructional scenario representative of such a workflow model.

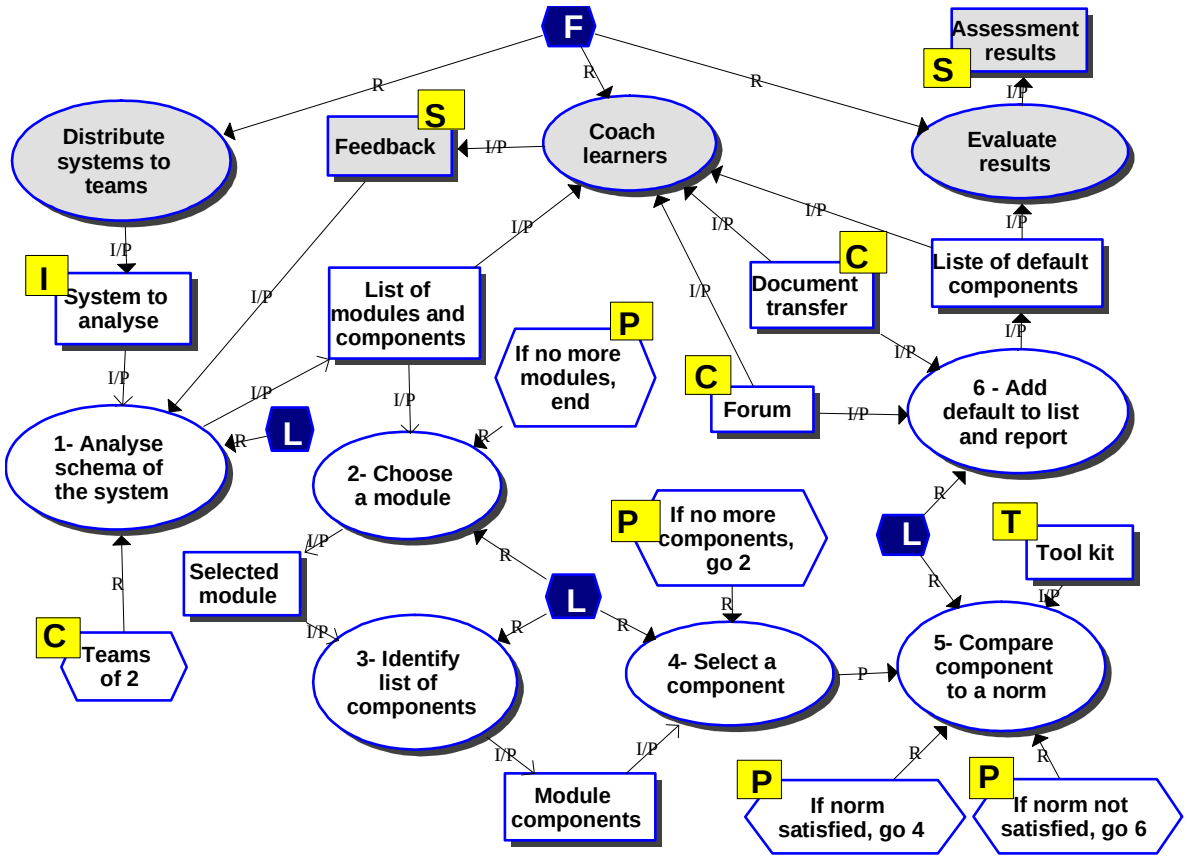

Figure 4– An example of a MISA scenario for learning to diagnose equipment failures

In the learning scenario subset (white ovals), learners (label **L**) perform 6 activities, starting with the analysis of an electronic system for troubleshooting. A collaboration rule (label C) states that they work in teams of 2. Progression rules (label P) define iterative cycles between activities until the complete electronic system has been analyzed. Through these cycles, each team of learners uses learning ob-

jects (label I) as inputs and produces intermediate outcomes, which finally results in a list of default components. Using an assistance scenario (grey ovals), facilitators (label F) start by distributing the system to be analyzed by the teams of learners, then providing feedback using a forum and document transfer, and finally providing assessment services to learners.

The instructional model encompasses five types of resources: instruments (documents/materials), tools/applications, services, locations (where learning is carried out) and communication means (such as "broadband", mail or face-to-face). These categories are expanded into sub-classes creating a complementary typology to the IEEE LOM Learning Resource typology[6]. In our definition, an instrument is the only type of resource that holds content. More precisely, they are associated to a sub-model in the Knowledge Model. We distinguish the "instrument" concept from the "learning materials" because they can, in general, be produced in different media formats. Usually, instruments are small pieces of information consulted or produced as a result of performing an activity and which, in turn, can be grouped and implemented in a one or more media formats (to increase accessibility) to create a certain type of learning material, such as a tutorial, handbook, guide etc... In particular, evaluation material, such as a questionnaire, exam or essay, is also associated to a knowledge sub-model and the target competencies are linked to the knowledge units in that sub-model. These competencies are the basis on which evaluation is developed and carried out.

The method MISA itself has been modeled using the MOT+ knowledge representation technique and tool. The relationship between MISA's tasks has been clearly and systematically represented using a process graph for each of the tasks. In the MISA documentation, this information is presented in the context table for each design

[6] See (IEEE 2002), Group Educational 5.2 Learning Resource type: exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self-assessment and lecture. Interested IMS-LD groups propose that this typology should be extended to include for example Unit of Learning and instructional methods.

element (DE). Table 2 presents this type of contextual information for the task "Define the instructional scenarios", which produces the DE 320 – Instructional scenarios. The list of DE source on the left, include some input information useful to the task that produces the DE 320, the list of DE to the right, uses information provided or produced in task 320.

Source				*Target*		
104	Target Populations			222-3	Learning Event Network	
212	Knowledge Model			224-3	Learning Unit Properties	I
214	Target Competencies			230-3	Material Production Orientation Properties	
220	Instructional Model Orientation Principles			240-3	Delivery Orientation Principles	
222	Learning Event Network	A	320	322	Properties of Each Learning Activity	A
224	Learning Unit Properties			330	Development Infrastructure	
230	Material Production Orientation Properties			340	Delivery Planning	
240	Delivery Orientation Principles			410	Content of the Learning Instruments	A
310	Learning Unit Content			420	Properties of the Instructional Instruments and Guides	

Table 2: A context model for an instructional design task in MISA

To support the propagation of data from one design task to the other, we have developed a web-based instructional engineering workbench, ADISA (Distributed Workshop for Engineering Training/Learning Systems). For each DE, the contextual information table uses labels A (automatic), S (source), or I (informative) to indicate which data propagation type is used in ADISA. Propagation is automatic when the data is directly used and necessary to carry out the task in ADISA. Data is displayed in the designer's interface when he starts the task. Propagation is semi-automatic when the data from the source need to be accepted by the designer before. Informative propagation means that the designer may consult some data in-

formation that might influence decisions for the task at hand.

The design documents of MISA can be edited in a flexible order, however according to data propagation rules, and can be modified, published in several stages, stored in archives, displayed on screen or printed. The data in the design documents are translated into a unified XML structure, allowing both online and offline work through an integrated web-based interface. It can be seen as a task map, allowing data propagation from one task interface to another, and also facilitating the information transfer to other systems. Other than supporting the data propagation between and among tasks and elements, ADISA supports the coordination of a group of experts, who plans and develops an instructional learning system, working both on and off-line.

3. Graphical Modeling of Learning Designs

In this section, we situate the MISA/MOT+ as an Educational Modeling Language (EML), followed by a presentation of the graphical symbolism integrated into the MOT+ graphical editor. Instructional designers will use this graphical representation language to build an IMS compliant Learning Design. Finally, we discuss the advantages of using the MOT+ graphical representation language and tool as well as new features to be added in order to become a fully compliant IMS-LD editor.

3.1 MISA/MOT+ as an Educational Modeling Language

In a study on Educational Modeling Languages, Rawlings et al. (2002) give the following definition: "*An EML is a semantic information model and binding, describing the content and process within a 'unit-of-learning' from a pedagogical perspective in order to support reuse and interoperability*".

According to this definition, MISA's specification of an Instruc-

tional Model is a kind of EML. The set of MOT+ models inherent in the Learning Event Network, plus the Instructional Scenarios of each Learning Unit, represented in a graphical way, can be directly compared to a semantic information model describing the content and processes of any unit-of-learning from an Instructional Engineering perspective. The translation of MOT+ models into XML files, automatically or by hand using an XML editor, makes possible interoperability and promotes reusability.

The MOT+ editor, which produces models like figure 3 and 4, has a built-in translator that produces an XML description of any such MOT+ graph. This translator has been used in the ADISA Web-based support system to propagate information from one design element to another (Paquette, et al. 2001). These XML files list the objects, links, sub-models, their properties and their interrelations. They do not constitute an IMS-LD XML binding, and a parser is under development to be added to the MOT+ tool, that can translate these XML structures into to standard machine-readable IMS-LD XML files.

3.2 A Graphical Language to Represent an IMS-LD Method Structure

When activating a Unit of Learning at runtime, the Method part of the XML file is central. This unique element and its sub-elements control the behavior of the Unit of Learning as a whole, coordinating the activities of the actors in their various roles and their use of resources.

As presented in the previous chapters, and displayed in Figure 5, the Method components, Plays, Acts and Role-parts, are all nested within each other. Plays are alternative scenarios run in parallel, while acts in a play are run in sequence. Within each act, role-parts are run in parallel, associating an actor's role to an activity (or to a more complex activity structure).

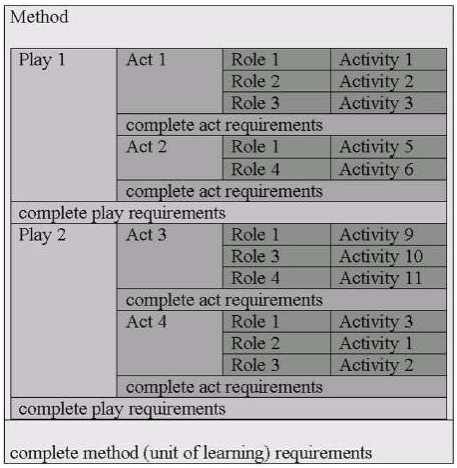

Figure 5- An IMS-LD Method

Because the MISA/MOT+ graphical representation system is generic, used for many kinds of models, such as e.g., representing domain ontologies or delivery process models, the MOT+ editor needs to be constrained in order to facilitate the modelling of IMS-LD compliant Units of Learning. To accommodate all the IMS-LD components, a set of graphical conventions have been specified and an IMS-LD XML parser for MOT+ is under development. Figure 6 displays some of the symbolism used.

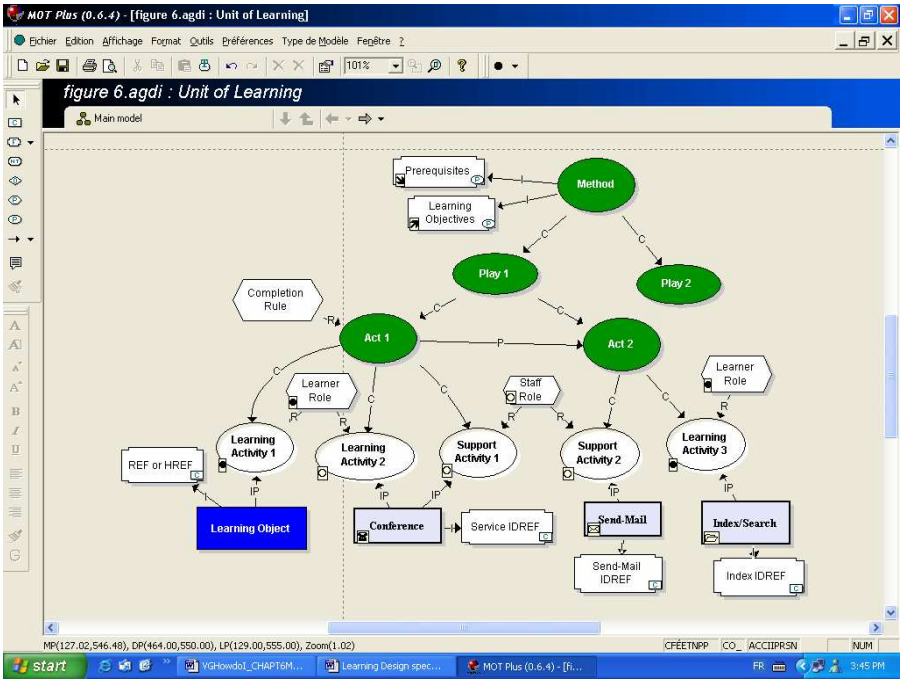

Figure 6– An equivalent MISA/MOT+ model to an IMS-LD example

Within MOT+, some combinations of specific graphic symbols, labels and links can be used to describing all the IMS-LD components and to produce a compliant XML document.

With the MOT+ IMS-LD adapted user-interface, the user will be presented with a Method model consisting of one Play, one Act and one Activity, which is the smallest possible structure for a Unit of Learning. All procedures, such as the Method, Plays, Act, Activities or Activity structures are represented as MOT+ procedures (ovals) and organized as a hierarchy using a composition links (C-link). To facilitate the interpretation and visualization of complex models, the activities in an act are embedded in a MOT+ sub-model, instead of being integrated into the main model as shown in figure 6. The precedence link, P-link, between acts illustrates a sequence of acts or activities. The absence of such links between activities denotes that they can be performed in any order (in parallel). Rules can be added

at any level, using a white hexagon symbol, e.g., completion rules at any level.

At the activity (or activity structure) level, role-parts are represented as the combination of a role R-linked to an activity or an activity-structure. A shadowed hexagon represents the role, associated by a responsibility R-link from the role to the activity or the activity structure. Icon-labels attached to the role symbol and on the activity symbol indicate whether it is a learner (black dot icon) or staff (white dot icon) role or learning or support activity.

Environments, containing learning objects and services are represented as concept objects (rectangles) and associated to activities through an input or product I/P-link, depending on whether they are used to carry out the activity (input), or produced (output) by performing the activity. Note that environments can be composed of many resources and services, which can be organized into a sub-model, using C links to indicate relationships. Different icon-labels distinguish content resources (white squares) from the three kinds of IMS-LD services: conference (telephone icon), email (letter icon) and index-search (folder icon). An internal or external reference can be associated to any resource using an instantiation I-link from the resource to the reference. The reference item is represented by a fact symbol (rectangle with cut angles). Learning Objectives and Prerequisites are represented by a fact symbol bearing a icon label in form of upward versus downward pointing arrows, as shown in figure 6. To respect the IMS specifications, the designer can only attach these symbols to the Method or to a Learning Activity.

At all levels of the learning design structure, time limit completion conditions can be defined using a white hexagon. If this symbol is absent, the parser interprets the completion condition as “user-choice”.

3.3 Using a MOT+ Editor

Graphical representational techniques and tools will free instructional designers from using XML editors and viewers in order to

consult either global or partial views of their design. Although well suited for software engineering purposes, UML graphs and diagrams, as proposed by the IMS Learning Design Best Practice and Implementation Guide[7], are not suited for instructional design, except maybe in very simple cases. Complex Units of Learning scenarios, especially those involving many actors, are not easily represented using UML graphs and activity diagrams. Moreover, it is important that all the IMS-LD components can be integrated using only one type of graphical model. This would greatly reduce the learning curve for designers to acquire a technique for constructing IMS compatible Learning Designs, which in turn would increases the possibility of interoperability and reusability.

The advantage of a graphical editor as compared to an XML editor is that designs can be structured and easily modified in an iterative manner, which is common practice for instructional designers when developing training courses and programs. An XML editor obliges the designer to declare all components of a Unit of Learning (Roles, Resources, and Activities), then to specify the Method structure and finally, to list all resource references. In the MOT+ editor, the designer proceeds by constructing the course structure (Method, Plays, Acts, Activities and Activity Structures), then adding environments with its learning objects and services as well as rules for progression and completion in an cyclic fashion. In this way, preliminary designs and milestones can be presented and validated by team members and clients, avoiding both costly and time consuming redesigns. Once consent is reached, the MOT+ editor allows the designer to save the Unit of Learning as a perfectly compliant IMS-LD XML document, ready to be used in a Content Packaging[8] tool, yet to be developed, or to be instantiated for a run in a compatible Learning Content Management System, such as Explor@2 or ATutor[9].

Many years of modeling courses and programs, for both universities and companies, have shown the MOT+ strength and user-

[7] http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html

[8] For single-user Units of Learning RELOAD http://www.reload.ac.uk/ex/ReloadSSv1.pdf

[9] Explor@2 demo http://lice.teluq.uquebec.ca and ATutor http://www.atutor.ca/

friendliness for non-computer professionals. Furthermore, the object oriented paradigm (Paquette 1996, 1999) distinguishing objects that represent facts, concepts, procedures and principles related by a standard set of links, is rooted in Instructional Design theories as well as in Information Sciences, and thus provides a strong basis as a notational language for learning designs.

4. An IMS-LD Case Study

In this last section, we will use the Versailles Experience (IMS-LD 2003) to develop and build an IMS-LD compliant Level A Unit of Learning using the MOT+ editor. We will then discuss the design method and tool used to build the model for this case.

4.1 The Versailles Narrative (extracted from IMS-LD, 2003)

The Versailles Experience is aimed at 14-16 year-old secondary school students. Participating schools organize students into six groups, one for each of the countries involved in negotiating the original Treaty of Versailles at the end of World War I: Great Britain, USA, Poland, France, Serbia and Italy. The design is based on collaborative learning and the duration is 4 to 6 weeks. The Unit of Learning has three main phases:

1) *A preparatory phase* in which students explore the content to find out what their role is, the context of their adopted country and agree on priorities and strategies for the forthcoming negotiation. In this pre-negotiation period participants in each school are organized into the six national negotiating teams, where each participating school is given six passwords - one for each country. These give access to the appropriate materials and a discussion group (dedicated conference) set up for each nation. Ahead of the actual negotiation, the tasks of the national teams are to:
   - become familiar with their country's objectives,
   - decide on their country’s priorities – what they most want and what they can concede,
   - become familiar with the objectives of the other countries,
   - identify possible negotiating strategies and agree the favored approach.

2) *The negotiation itself.* For the Negotiation Day, there is a main negotiation forum with a conference Chair, but there are also 'side rooms' for each pair of countries to hold private discussions. These are set up as dedicated conferences with appropriate access provided for each team. When agreements are reached during negotiations, they are sent to a person playing the role of a Recorder who posts them on a 'Results Board'. Participants have access to the results at any time. Once the negotiations are completed, or at a given time towards the end of the day, participants are encouraged to review the outcomes of the day.
3) *A post-negotiation period* offer the students the opportunity to disseminate what they have learned in the form web-based materials presenting national perceptions of what the treaty meant to each of the participating nations. In this last phase, students reflect on what they have learned, writing it up from the point of view of what the outcomes mean for their adopted countries. This involves both face-to face activities in each school as well as using the country team forums. These are then translated into Web pages and posted under a preset page for each country. Students then review their collective postings.

4.2 A MOT+ representation of the Versailles Case

We have build a MOT+ model of that learning unit, using the graphical conventions presented in the previous chapter. Because of the complexity of this learning situation, we need to use embedded activity structures (labeled by a bulls-eye icon) using MOT+ sub-models.

The main model presents the Unit of Learning structure, the LD Method. The method is composed of one play divided into eight sequential acts as shown in figure 7. Each act is described in a sub-model. Act 1, 2, 4, 5, 7 and 8 are simple Acts that are not decomposed further, that is they do not contain embedded activity-structures, just simple role-plays where a role performs a single learning of staff activity.

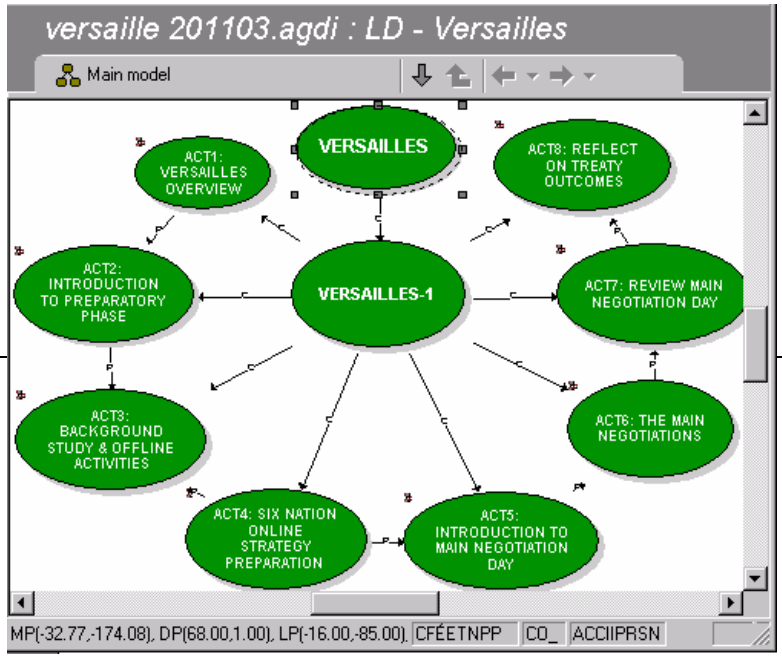

Figure 7– The Versailles' main model

Act 4 is an example of a simple act as shown in figure 8. This sub-model displays one central activity structure performed by two staff roles, a teacher and an expert. The activity structure is composed of six learners' roles and their corresponding learning activities, one for each country. Each national team (hexagon with Country Name) uses a private conference (rectangle with telephone label) to establish the country's negotiation strategy. Results can be accessed by all.

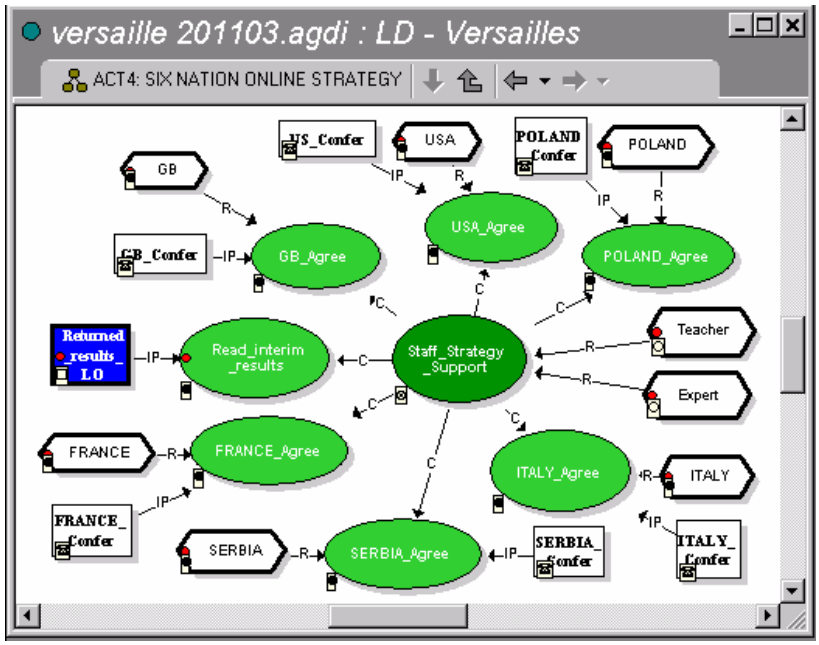

Figure 8– A Sub-model for Act 4: SIX NATION ONLINE STRATEGY

Figure 9 presents the main model for Act 6 covering the activities on the negotiation day between the six teams. At the center, there is an activity structure, “Main_Negociate”, which uses an environment composed of a general conference in which there are two conference activities, actually indicating user-rights for the conference (see IMS Information Model 3.1.11) : “moderate” played by a staff person called “Chair”, and “participate” played by all learners, plus a teacher and an expert.

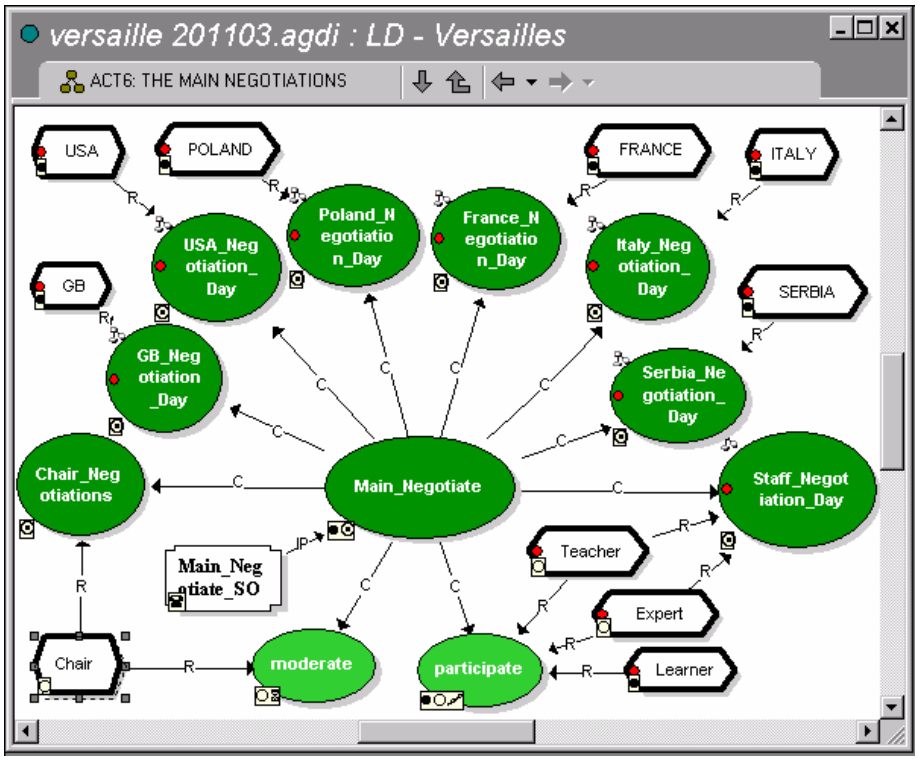

Figure 9 – A Sub-model for Versailles’ Act 6: THE MAIN NEGOCIATIONS

The central activity structure is further decomposed into 8 other activity structures; all performed in parallel, shown by the absence of

precedence (P) links. Six of them correspond to each national team of learners, associated to corresponding role-parts in the activity structure, each developed in a sub-model constituting a third level of models (this is shown by the little model icon on upper left of the oval). There are two more activity structures: “Chair_Negociations” performed by the Chair role, and “Staff_Negociation_Day” performed by Teacher and Expert roles.

Figure 10 presents one of the third level activity structures, the one where the Great Britain (GB) team is involved. The lower part of the figure shows that it is decomposed into five learning activities where the GB team is involved in negotiations with each of the five other teams. For this, specialized conferences are open in the environment and each activity produces five corresponding agreements (dark blue rectangles = products).

The upper part of the model in figure 10 illustrates the exchange of information between GB learners and staff. There are three such learner activities, one where GB learners send the results of their negotiation using an email service, another one where a GB-learner, taking the role of a Recorder, receives results in a mailbox and does some Web editing, and a last one where this aggregated result is returned to GB learners and staff. Note that since GB learners are associated to the central activity structure, it is not necessary to repeat this association for the other learner activities. By default, it is inherited through the C link.

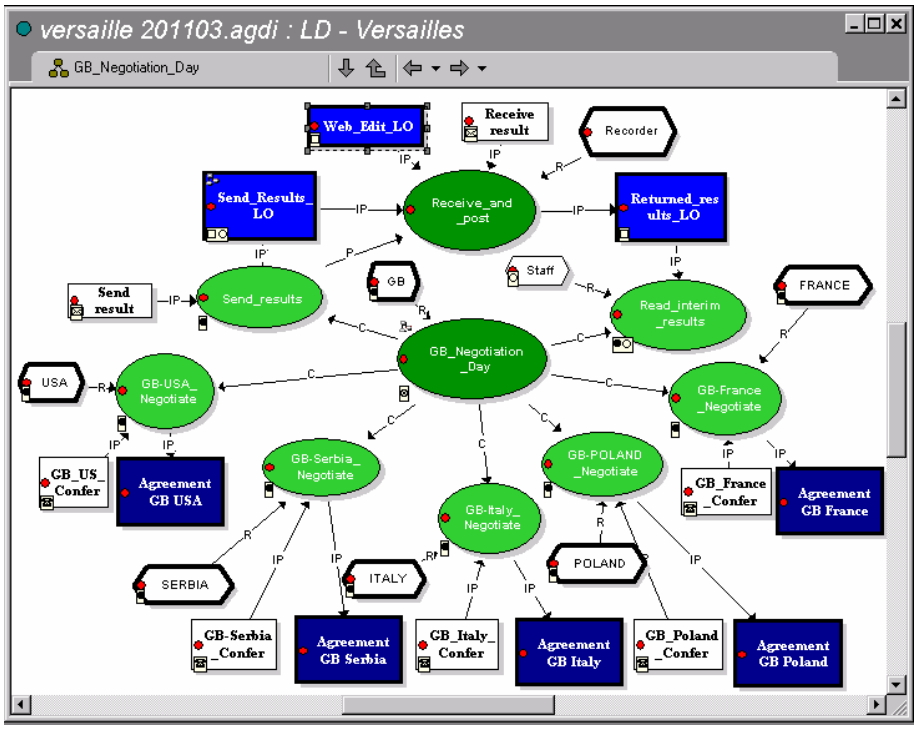

Figure 10 – A Sub-sub-model for Activity structure: GB_NEGOCIATION_DAY

4.3 Discussion of the case

This case is probably one of the most complex Unit of Learning scenarios that have been developed so far. In the classroom, a teacher would probably spend many hours explaining it to the learners. Collaborative scenarios like the Versailles example may have great learning benefits, but are difficult to implement in a classroom and even more so in network-based environments. The advantage of providing a structural graphical model is that it can also serve as a task guide for both students and teachers, thus avoiding lengthy and repeated explanations.

The modeling of learning designs brings the greatest benefits, when the learning situations involve multiple roles, where the activities are

not sequential, and where their results are reinvested in other activities. The process of building a model helps the designer to clarify his ideas and communicate them to the learners, whether in a class or acting as an on-line coach.

But there is more to it. If machine support is expected in a computerized networked environment, it is essential to formalize the flow of activities and precisely identify the actors, their roles as well as the resources used or produced in the environment. Once the graphical formalization is done, it can automatically be translated into IMS-LD XML machine-readable code, without the direct intervention from the designer.

Conclusion

The adoption, at the end of 2002, by IMS of the OUNL Educational modeling language as the basis for a standard specification, is great progress. It enables knowledge-based Instructional Engineering methods, like MISA, to produce learning designs that can potentially be read by any compliant LCMS, as is discussed in the following chapters describing case studies.

We have shown that the IMS-LD specification and the MISA method complement each other, by proposing an instructional engineering method in six phases, specifying four axes through the elaboration of knowledge and competency model, a pedagogical model, as well as resulting learning material and delivery models. The IMS-LD specification provides a standardized formal and machine-readable representation of a learning design, whereas MISA proposes a systemic and systematic method to design and implement such learning designs. The MOT+ graphical editor, used to implement the MISA method, also appears as a promising alternative to UML modeling, mainly because it is rooted in instructional design theory and has been built with education and training applications in mind.

In 2004, we are completing the integration of IMS-LD related tools

in the eduSource[10] Suite of Tools application, which already contains an implementation of standards for learning objects repository interoperability. In the five-year term of the LORNET[11] project, we will be working to extend the LD specification to more general function or workflow models (Paquette & Rosca 2002), and to adapt our Explor@2 delivery system to fully exploit the multi-actor concept claimed by IMS-LD specification.

On a larger scale, we believe that international standardization efforts should focus on the very important question of the association of knowledge and competencies to the IMS-LD method components. In a Semantic Web perspective, this is an essential task where strong international collaboration is needed.

References

Ausubel, D. P. (1968). Educational Psychology: A Cognitive View. New York: Holt, Rinehart and Winston.

Berners-Lee, T., Hendler J., Lassila,O. (2000) The Semantic Web. Scientific American, May 2001, Feature article.

Breuker J. and Van de Velde W. (1994) CommonKads Library for Expertise Modelling. IOS Press, Amsterdam, 360 pages, 1994.

Breuker, J., Muntjewerff, A., and Bredewej, B. (1999). Ontological modelling for designing educational systems. In *Proceedings of the AI-ED 99 Workshop on Ontologies for Educational Systems*, Le Mans, France. IOS Press.

Bruner, J. (1966).. Toward a Theory of Instruction. Cambridge, MA, Belknap Press/Harvard University Press.

Collins, A & Stevens, A.L. (1983). A Cognitive Theory of Inquiry

[10] eduSource is a large Canadian project that it implementing many IMS specifications. In this project, our group is responsible for the integration of the open source software infrastructure. To see the LD work in this project, consult the eduSource use cases document at www.edusource.ca.

[11] LORNET is a new Canadian Research Network on a five year mission. To know more about this project, consult www.lornet.org.

Teaching. In Reigeluth, C.M. (ed.) Instructional Design Theories and Models: An Overview of their Current Status. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Dewey J. Psychology and social practice. (1900)The psychological Review. 1900, 7, pp 105-124

Gagne, R. N. (1970). .The conditions of learning. New York: Holt, Rinehart & Winston.

IMS-LD (2003). *IMS Learning Design. Information Model, Best Practice and Implementation Guide, Binding document, Schemas*. Retrieved October 3, 2003, from http://www.imsglobal.org/learningdesign/index.cfm

IEEE (2002). Draft Standards for Learning Object Metadata Learning Technologies Standards Committee of the IEEE. IEEE 1484.12.1 - 2002 http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf

IEEE LTSC (1999) Learning Object Metadata, Draft Document v3.6). IEEE Learning Technology Standards Committee (LTSC)., 5 September 1999 http://ltsc.ieee.org/doc/wg12/LOM3.6.html

Koper R. (2002). Modeling units of study from a pedagogical perspective – The pedagogical metamodel behind EML http://www.eml.ou.nl/introduction/articles.htm Last consulted, March..

Landa, L.N. (1976). Instructional Regulation and Control: Cybernetics, Algorithmization and Heuristics in Education. Englewood Cliffs, N.J.: Educational Technology Publications.

Le Moigne, J.-L. (1995). Les épistémologies constructivistes, ed. PUF Que sais-je? Paris.

Merrill, M.D., Reigeluth, C., & Faust, G. (1979). The instructional quality profile: Curriculum evaluation and design tool. In H. O'Neil (ed.), Procedures for Instructional Systems Development. New York: Academic Press.

Merrill M.D. Principles of Instructional Design (1994) Educational Technology Publications, Englewood Cliffs, New Jersey, 465 pages.

Paquette G. (1996) La modélisation par objets typés: une méthode de représentation pour les systèmes d'apprentissage et d'aide

a la tâche. Sciences et techniques éducatives , France, avril 1996

Paquette, G. (1999) Meta-knowledge Representation for Learning Scenarios Engineering. Proceedings of AI-Ed'99 in AI and Education, open learning environments, S. Lajoie et M. Vivet (Eds), IOS Press, 1999.

Paquette G. (2001a) TeleLearning Systems Engineering – Towards a new ISD model. Journal of Structural Learning 14, pp. 1-35,

Paquette,G., Rosca, I. , De la Teja, I., Léonard, M., and Lundgren-Cayrol, K. (2001). Web-based Support for the Instructional Engineering of E-learning Systems, WebNet'01 Conference, Orlando.

Paquette, G. (2002a). L'ingénierie pédagogique, pour construire l'apprentissage en réseau Presses de l'Université du Québec, 457 pages

Paquette, G. (2002b) La modélisation des connaissances et des compétences, pour concevoir et apprendrePresses de l'Université du Québec, 352 pages

Paquette G. and I. Rosca. (2002) Organic Aggregation of Knowledge Objects in Educational Systems, Canadian Journal of Learning Technologies, Volume 28-3, Fall 2002, pp. 11-26

Paquette (2003) Paquette, G. Instructional Engineering for Network-Based Learning. Pfeiffer/Wiley Publishing Co, 262 pages.

Rawlings, A., P. Van Rosmalen, R. Koper, M. Rodriguez-Artacho, P. Lefrere. (2002). Survey of Educational Modelling Languages (EMLs), version 1, September 19th, CES/ISSS

Reigeluth C. (1983). Instructional Theories in Action: Lessons Illustrating Selected Theories and Models. Reigeluth C. (Ed) Hillsdale, NJ: Lawrence Earlbaum, 487pp.

Reigeluth, C.M. and Rodgers, C.A. (1980) The Elaboration Theory of Instruction: Prescription for Task Analysis and Design, NSPI Journal.

Romiszowski A. J..(1981). Designing Instructional Systems. Kogan Page London/Nichols Publishing, New York, 415 pages.

Scandura, J.M. (1973). Structural Learning I: Theory and Research. London: Gordon & Breach.

Simon, H. A., (1973), The Organization of Complex Systems in: Hi-

erarchy Theory. The Challenge of Complex Systems, H. H. Pattee (ed.), (George Braziller, New York), pp. 1-27.

Skinner, B.F. (1974) About behaviorism. New York: Knopf.

Spector J.M., Polson M.C., Muraida D.J. (Eds) (1993). Educational Technology Publications, Englewood Cliffs, New Jersey, 364 pages, 1993

Wiley D.A. (2002). *Connecting learning objects to Instructional design theory: a definition, a methaphor, and a taxonomy*. In Wiley (Ed) The Instructional Use of Learning Objects. Agency for Instructional Technology and Association for Educational Communications of Technology, Bloomington, Indiana, , 281 pages.