E^{2}ML - Educational Environment Modeling Language

Luca Botturi luca.botturi@lu.unisi.ch
NewMinE Lab
Università della Svizzera italiana
Via Buffi 13, CH-6900, Lugano

Abstract. *E^{2}ML is a modeling language specifically developed for the design of educational environments. It models the goals, requirement and design of the teaching and learning activity. Its core parts are a verbal statement and visual mapping of goals and a visual UML-like representation of the learning activity. E^{2}ML can be profitably integrated with Instructional Design models in order to enhance the communication within design team and the elicitation of requirements, and to provide a successful integration of new media in the learning experience through a well-shaped and documented design process.*

Introduction: New Challenges in the Design of Education

E-learning has brought a cultural shift in the way we conceive teaching and learning, and in the everyday practice of instructors. New media and networking have made available several tools for education, and these in turn make possible a number of new educational strategies based on distributed team working, on asynchronous online communication, etc. Instructional Design (ID) is the discipline concerned with research and theory about instructional strategies and the process for developing and implementing those strategies (Smith & Ragan 1993). ID models are tools that can be used for structuring a correct design process and for checking the quality of the final product. During the last years, they have been revised and updated in order to fit to e-learning design, but still the design of education remains a hard task in any context.

In this paper we present E^{2}ML, a semi-formal representation language for educational environments that can be profitably exploited and integrated with other models in instructional design. The first paragraph will provide a short state of the art in ID, while the following ones will introduce E^{2}ML with definitions and examples. Finally, conclusions and outlooks are reported.

Instructional Design: Where Are We?

Gustafson and Branch have categorized ID models in three main categories (Gustafson & Branch 1991, Gustafson & Branch 1997).

1. *Classroom-oriented models* represent the greatest part of ID models. They are used for designing one or a few hours of instruction, mainly in the traditional classroom setting. They assume that one instructor is at work, the exemplary situation being that of a schoolteacher preparing a lesson or an activity. These models serve as checklist for designers, guiding them from the requirement analysis to the final evaluation of results before redesign, like a specialized project management red-thread. The basic model is ADDIE (Analyze, Design, Develop, Implement, Evaluate). Other models are Gagne's Nine Events of Instruction (Gagné 1987; Gagné et Al. 1992) and ASSURE (Heinrich et Al. 1983).

2. *Product-oriented models* are aimed at designing effective learning materials. These models have found new relevance and new challenges with the introduction of new technologies, of multimedia and distance communication means. Like other hypermedia design models, they are concerned with a team at work. One of the most interesting models is CADMOS-D, a specialized UML notation for educational applications (Retalis et Al. 2002, Psaromiligkos & Retalis 2002)

3. *System-oriented models* are developed for a team designing a great amount of instructional units, such as a course or a curriculum. They are usually more technical than classroom-oriented models and also guide the

design team through a well-shaped process to the achievement of the expected result. The main referenced models are by Dick and Carey (Dick & Carey 1996) and by Ragan and Smith (Smith & Ragan 1993).

On the other side, several organizations in the technical world such as IEEE, ADL and IMS, are investing much in the development of metadata standards for Learning Objects (see e.g. LTSC LOM) and for the description of whole learning environments (noticeably IMS Learning Design, see IMS 2003a and IMS 2003b). Although these developments do not belong to the field of ID, their advancement will influence it in a probably near future.

What is E^{2}ML?

E^{2}ML – Educational Environment Modeling Language is proposed as a blueprint for educational environment modeling. As conceptual framework, E^{2}ML is composed by an explicit definition of the learning process and of the educational activities; as a formal modeling language, it defines UML-like representation syntax.

Differently from the greatest part of ID models, E^{2}ML was developed for representing *the educational activity* being designed, and not the process of design itself or the support materials developed. This makes it profitably integrable with other models.

When is E^{2}ML useful?

Instructional design models are structured design methods, and as such require an overhead cost. A structured design is cost-effective only when the complexity of the designed object is high.

E^{2}ML was developed for supporting educational design in technology-rich contexts, and it is useful when an interdisciplinary team is at work on a course that exploits different media and different kinds of educational activities and settings. In particular, it addresses the following issues:

1. The subject of e-learning design is often an interdisciplinary team, which may include subject experts, media designers, technicians, tutors, psychologists, etc. External partners too may have a role in delivery or supply (see e.g. Pigni 2002, Ardizzone & Oliveto 2002). E^{2}ML can *ease and enhance communication* in the design team and with external partners. Moreover, an E^{2}ML representation of the educational activity could be the basis for a project management approach to course development, structured according to other classroom-oriented or system-oriented ID models.
2. The definition of requirements for the tools to be used in a specific e-learning environment is a delicate issue, as it often concerns great investments, both in the case of off-the-shelf solutions and in that of custom application development. E^{2}ML is a tool for systematically defining and *expressing the educational requirements* of software applications, thus providing a first input to product-oriented ID models.
3. Setting up an e-learning environment is great economic effort (developing content, digitizing, buying and customizing applications, training tutors, etc.), and should be balanced by an adequate return on investment. E^{2}ML can support *static quality assessment* through the expression of formal features. This can help controlling the quality of the learning experience at design time and distinguishing quality-critical applications or content from optional nice-to-have features and optimizing investments.
4. Given the complexity of e-learning environments and the uniqueness of each class and of each learner, unexpected learning outcomes may rise. E^{2}ML modeling may be used as a *diagnostic tool* for identifying relevant issues and for figuring out viable redesign solutions.

The E^{2}ML modeling of a learning environment requires four main activities: the expression of goals (goal statement and goal mapping); the compilation of resource lists (roles & actors, locations, tools); the definition of action diagrams and finally the creation of overview diagrams (course breakdown statement – CBS, dependencies, timeline).

These activities are described in the following paragraphs in their standard form. Nevertheless, they can be adapted (simplified or detailed) to the needs of any specific educational context or design team. Moreover, their presentation in this paper does not coincide with their actual performance in a real design process – experience proved that most of the time the activities run in parallel and interact with each other.

Goal Definition

Goal Statement

As all ID models indicate, the learning goals should be clearly stated as the expected outcome of the learning process on the learners' side. E^2ML provides a standard format for a goal list, where goals are described by the following elements: a unique identifier tag, the statement, the target (who should achieve the goal), the stakeholder (who expressed that goal, for whom it is important as outcome of the learning activity) and assessment (how you will assess that the goal was achieved). Moreover, each goal can be given an importance score: this may help ranking them. The importance score can be calculated as generic value (referring e.g. to the instructor's perception, or to students feedback) or as (balanced) average of the individual importance assigned by the different stakeholders. The example in the table above (like all the others in this paper) is taken from a course on institutional communication held in Lugano to 170 freshmen in Summer Semester 2002 (see Botturi et Al. 2003).

TAG	STATEMENT	TARGET	STAKE HOLDER	ASSESSMENT	IMP
G1	Acquire general concepts of organizational theory for institutions and institutional communication	Students	Instructor	Exam (questions)	4
G2	Acquire framework for the categorization of different institutions	Students	Instructor	Report + Exam	5
G3	Describe, classify and compare institutions	Students	Instructor	Report (R1 + R2)	3
G4	Analyse complex institutions and figure out possible solutions to critical issues	Students	Instructor, students	Exam (case study)	3
G5	Recall examples (and „direct experience") of institutions and best practices in institutional communication	Students	Instructor	Exam (do examples)	5

Goal Mapping

Goals should then be mapped on a (possibly visual) representation of the learning dynamic in its general structure (e.g. Bloom's taxonomy, Lonergan's learning dynamic, etc.), or of the discipline itself, as a knowledge structure (e.g. a concept map of Linguistics, etc.)

E^2ML offers a model for representing high-level goals by stating their type (cognitive, psycho-motorial, affective, etc., taken from Gagné's types of learning outcomes), their scope (recall, use or find; taken from Merrill 1983) and level (taken from Lonergan 1990). The model is sketched in Figure 1.

A visual mapping of goals on the learning dynamic is useful for during-course and post-course assessment: learning problems (e.g. difficulties in a certain step in the learning dynamic) or unexpected learning outcomes (e.g. a goal is not achieved) can be better understood with this tool, as explained later on.
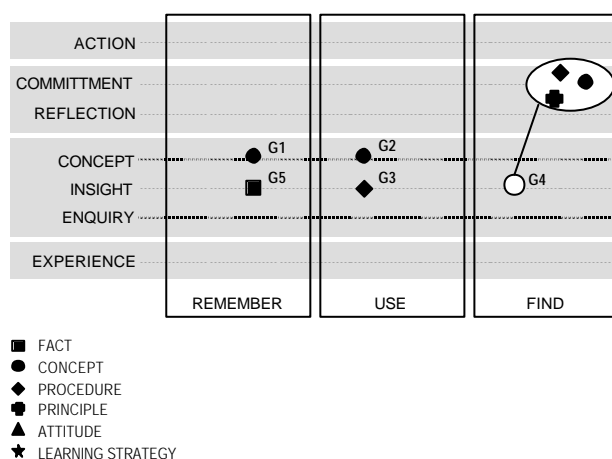

Figure 1 - Example of visual goal mapping

Resource Lists

The second step in E^2ML modeling is writing three lists that describe the available resources. For reasons of space we will not include here examples of these lists.

1. ROLE AND ACTOR LIST: it is a description of the different functional roles that the persons involved in the course enactment may cover. Each role is described with a unique identifier tag, the role name, a description of the functions, the number of persons covering that role and eventually a list of their names. The role list is a support in controlling that every activity during the course has the proper profiles for completing it. The matching of roles and actors is also useful to check consistency between the person's profile and his/her assignments and to avoid work overload (specially if this list is used together with the timeline – see later).
2. LOCATION LIST: it is a description of the physical places that can be used for hosting the educational activities. Each is described with a unique identifier tag, its name, a description and a list of the available facilities. The location list is useful as a checklist to verify that every activity can happen at the proper place, and to have an overview of eventual reservations for spaces or facilities.
3. TOOL LIST: it is a description of the technology-dependent tools that are exploited in the environment. Each tool is described with a unique identifier tag, its name, a description (e.g. online website at URL, a book, a CD-ROM, etc.), notes on its availability, the reference of the person in charge of managing it (especially when he/she does not belong to the design team) and the contact information of the technical support. The tools list is a useful reference for the implementation status, and can be used for project coordination. The bigger the number of tools, the more it is useful.

Action diagrams

Along with goal statement and mapping, action diagrams are the core part of E^2ML modeling. The E^2ML framework is in fact centered on a representation of the TDEE as a system of educational activities (or actions). Actions are performed by actors with specific roles, exploit location and tools, and are aimed at specific goals or sub-goals.

Action Structure

An action is the minimal unit of the educational environment, intended as a *unity of purpose* (e.g. write a report, complete an exercise) *for a defined acting subject* (e.g. a single learner, a group, etc…) .

An action can be split in several sub-actions according to the time and/or space unity criterion in the specific setting (e.g. a single lecture, a videoconference). This second distinction (time/space) should of course match with the previous one (goal/subject)[1].
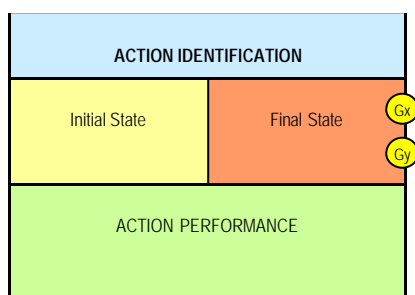

Figure 2 - General structure for an action in E^2ML
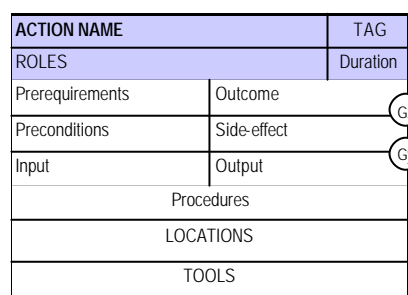

Figure 3 - Complete E^2ML action diagram

The general schema for the representation of an action is represented in Figure 2. The upper part of the diagram contains the proper identification for the action, i.e. its name and the involved roles. The middle-left area describes the initial state (necessary and sufficient conditions) for the actualization of the action, while the middle-right area describes the (desired) final state after the action performance. Finally, the lower part of the diagram contains a description of the actual performance (procedures), including the involved locations and tools. The circles on the right-hand side represent a reference to the goal list, and indicate what goals are addressed by the action. The complete E^2ML action diagram is actually more detailed, as in Figure 3.

---

[1] These considerations should be taken into account for selecting the granularity of the E^2ML representation. Generally, granularity should be adequate to the specific project or design context.

For reasons of space, we cannot provide here the explicit definition for all the descriptors in the diagram, which may nevertheless become clear confronting the example in Figure 4.

Case Studies In-Depth Analysis		R1
Students (all, single)		20h
Being able to use the institution description schema + can apply organizational theory concepts	Complete understanding of one institution	G3 G4
Using email + Using MS Word (basics)	-	G5
One online case study	Final Report (10 p.)	
View all materials of a case study's and analyze it with the course concepts. Ask the A for questions. Write the report following the report guidelines. (20h)		
[anywhere]		
Online case studies (CS)		
Syllabus		

Figure 4 - Example of E²ML action diagram

Actions can be *optional* or *compulsory*. A compulsory action is expected (or it is sensible to expect) that all learners perform it. This can be guaranteed by rules, assessment systems, etc. (e.g. attend the weekly lecture). Optional actions are included in the design, but learners are free to perform it, or it is unlikely that they all do that (e.g. take some time for revising notes every week, or any optional reading). Optional actions are represented like other actions, except that the external box border is dotted.

Relationships

At this level, actions can have two kinds of relationships among them:
1. INHERITANCE (triangle-end arrows in Figure 5). Action diagrams can represent action instances or action types (such as abstract and instance classes in UML). Action instances are single actions, identifiable as a defined event in the educational environment. Action types are general descriptions of a specific *kind* of action (e.g. a class discussion); as such types can be used as a model or a blueprint for defining instance actions. Action type diagrams are used to define common patterns of action that can be *inherited* by action instances (or other action types). Moreover, they can be exploited as general action models for reuse.
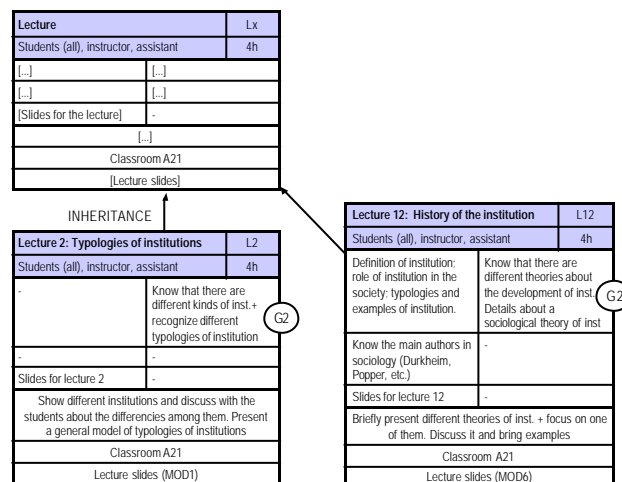

Figure 5 - An example of inheritance

2. AGGREGATION (diamond-end arrows in Figure 6). It indicates that one action is composed by two or more sub-actions. "Part actions" represent a sort of more detailed view of the "whole action", that can be divided e.g. in a sequence of smaller actions. Aggregated actions (parts) have the same goals or a subset of goals as the main action.
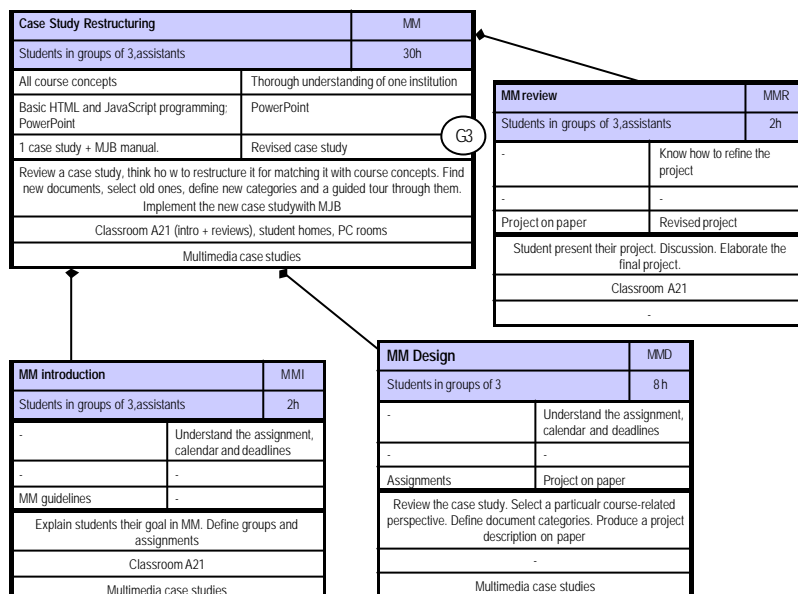

Case Study Restructuring		MM
Students in groups of 3,assistants		30h
All course concepts	Thorough understanding of one institution	
Basic HTML and JavaScript programming; PowerPoint	PowerPoint	
1 case study + MJB manual.	Revised case study	
Review a case study, think how to restructure it for matching it with course concepts. Find new documents, select old ones, define new categories and a guided tour through them. Implement the new case study with MJB		
Classroom A21 (intro + reviews), student homes, PC rooms		
Multimedia case studies		

MM review		MMR
Students in groups of 3,assistants		2h
-	Know how to refine the project	
-	-	
Project on paper	Revised project	
Student present their project. Discussion. Elaborate the final project.		
Classroom A21		
-		

MM introduction		MMI
Students in groups of 3,assistants		2h
-	Understand the assignment, calendar and deadlines	
-	-	
MM guidelines	-	
Explain students their goal in MM. Define groups and assignments		
Classroom A21		
Multimedia case studies		

MM Design		MMD
Students in groups of 3		8 h
-	Understand the assignment, calendar and deadlines	
-	-	
Assignments	Project on paper	
Review the case study. Select a particualr course-related perspective. Define document categories. Produce a project description on paper		
-		
Multimedia case studies		

Figure 6 - An example of aggregation

Overview diagrams

Overview diagrams are designed in order to provide the development team with a synthetic view of the whole learning environment. In order to improve legibility, actions can be represented here with a simplified diagram (a simple box containing the action's tag).

Course Breakdown Statement (CBS)

Project management best practices suggest the production of a list of all the task and subtasks in which a project is articulated. The same can be useful in course design. E^{2}ML defines the Course Breakdown Statement (CBS) as a list of all action instances that indicates, for each action, the tag, name, roles, locations, tools and duration descriptors.

The dependencies diagram

The dependencies diagram (see the example in Figure 7) represents all action instances and the relationships among them. The represented relationships are of three types:
1. LEARNING PRE-REQUIREMENT (circle-end arrows). A learning pre-requirement relationship indicates that the first action provides a (the) learning outcome that is the pre-requirement for the second action (e.g. a lecture provides concepts for the following analysis work).
2. INPUT/OUTPUT (simple-end arrows). An input/output relationship indicates that the first action produces as material output some artifact that is required as material input for the second action (e.g. a group-work activity produces a presentation which will be shown during the following class discussion).
3. AGGREGATION (see above, represented by nesting part actions in the whole action box).

In order to improve legibility, actions can be grouped into *plays* (rounded gray boxes in the Figure)
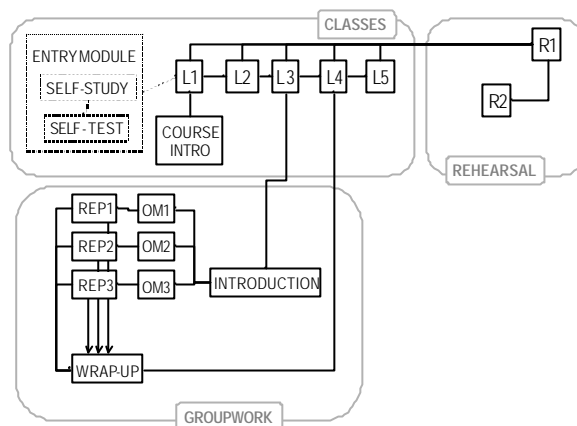
Figure 7 - An example of E^{2}ML dependency diagram

The dependencies diagram is useful for representing all the relationships existing among actions, and is the basis for a correct design of the timeline. During the course enactment, it is useful to figure out repetitions or new actions, as it allows identifying where learning or material problems may occur due e.g. to the failure of an action.

The timeline

The timeline is a visualization of the course calendar, and provides an overview of the flow of educational activities during the course time span. Like a Gantt chart, the timeline provides an adequate time grid (e.g. days, weeks or months) on which all action instances are represented. The whole period can be divided into phases. Specific dates (or other information) can be annotated beside actions. Eventually, UML notation for flowchart-like representation can be included (forks, joins, splits with conditions). An example is presented in Figure 8.

The timeline corresponds to the effective planning for the execution of actions. A first validation check should consider if it actually preserves the learning pre-requirement and input/output relationships drawn in the dependencies diagram. During the course enactment, it is useful for discussing the course progress. Moreover it can be used for planning application or content development that happens during the course.
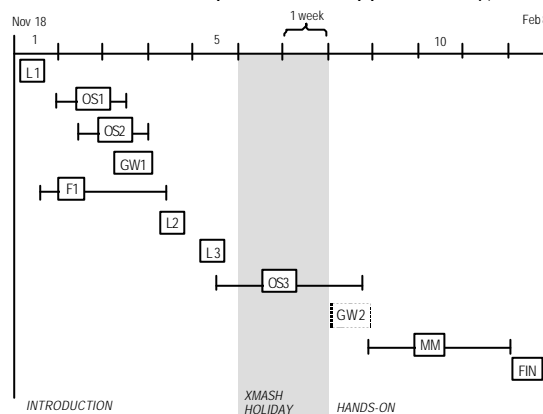

Figure 8 - An example of E^{2}ML timeline

Conclusions

E^{2}ML is a semi-formal language for representing educational environments. It is intended to be a new tool to fit into existing Id models. Its specific feature is to represent the designed educational environment in a semi-formal way. It can be exploited in instructional design as a mean for expressing the output of the design phase, and for creating a common understanding within the design team. Its simple visualization makes it a flexible tool for educators, and

does not require any technical competencies. E^2ML is a tool, which we hope can help instructional designers, teachers and IT developers to improve collaboration and to better express their professional competence in education.

A short remark should be made concerning compatibility with standards. The E^2ML model is compliant with IMS Learning Design for the representation of educational activities and methods and with IMS RDCEO (IMS 2003b) for expressing goals.

Current State & Outlooks

Currently, some design activities are carried out at the University of Lugano taking advantage E^2ML with good results on courses of Logics and Institutional Communication. Case studies will be published in the next months.

E^2ML is exploited as tool for working with professors (in the role of content experts), for expressing the educational requirements of the environment (through goal statement and goal mapping). Moreover, it supports a consistency check between goals and design. During the course, the E^2ML timeline serves as checklist that all materials are ready at the right mo ment. In case a lecture fails, or that any problem does not allow the completion of a particular activity, the dependency diagram is a support in quickly identifying possible compensation activities. After the course and the evaluation phase, the result may prove that some goal was not achieved, or that an unexpected learning outcome occurred. Working on the goal mapping it is possible to move backward and identifying what actions and tools were concerned with the achievement of that goal and start there the redesign phase.

The first step for a further development of E^2ML is the refinement of the language and of the representation diagrams. Moreover, the validation through practice and case studies will be the proof of its usefulness.

The development of an application that supports E^2ML design is currently in progress (as an add-in for Rational Rose$^{TM)}$. The application will also automatically produce a Learning Design compliant XML representation of the learning environment.

References

Smith, P.L. & Ragan, T.J. (1999). *Instructional Design (2nd ed.)*, Wiley & Sons Publishers.

Gustafson, K.L., & Branch, R.M. (1991). *Survey of Instructional Design Models*, ERIC report

Gustafson, K.L. & Branch, R.M. (1997). *Revisioning Models of Instructional Development*. In: *Educational Technology Research and Development*, vol 45, no. 3., p. 73-89

Gagné, R. (1987). *Instructional Technology Foundations*, Erlbaum Associated Press, NJ.

Gagné, R., Briggs, R & Wager, W. (1992). *Principles of Instructional Design*, TX: HBJ College Publishers

Heinrich, R., Molenda, M. & Russell, J. (1983). *Instructional Media and new Technologies*. Macmillan, New York

Retalis, S., Papasalouros, A. & Skordalakis, M. (2002). *Towards a Generic Conceptual Design Meta-Model for Web-Based Educational Applications*, 2nd IWWOST, Malaga, Spain

Psaromiligkos, Y. & Retalis, S. (2002). *Configuration Management for Web-Based Instructional Systems*, 2nd IWWOST, Malaga, Spain

Dick, W., Carey, W. & Carey, L. (1996). *The Systematic Design of Instruction* (4th ed.). New York: Harper Collins College Publishers.

Pigni, F. (2002). *FAD ed eLearning: il Modello DECUS dell'Università Cattolica*. In: Cep@d, *eLearning: Ricerca e Modelli Operativi – L'Esperienza dell'Università Cattolica*, Vita e Pensiero, Milano

Ardizzone, P. & Oliveto, B. (2002). *Corso a Distanza in Attività Turistiche e di Valorizzazione Culturale del Territorio*. In: Cep@d, *eLearning: Ricerca e Modelli Operativi – L'Esperienza dell'Università Cattolica*, Vita e Pensiero, Milano

Botturi, L., Inglese, T. & Rozalèn, S. (2003). *Design and Trust as Key Factor in Elearning*. AACE, SITE 2003, Albuquerque, New Mexico.

Lonergan, B. (1993). *Understanding and Being: The Halifax Lectures on Insight* (edited by E.A. Morelli and M.D. Morelli, University of Toronto Press, Toronto

IMS Consortium (2003a). *Learning Design Specification*, available online at learningnetworks.org

IMS Consortium (2003b). *Reusable Definition of Competence or Educational Objective Specification*, available online at learningnetworks.org

Merrill, M.D. (1983). *Component Display Theory*. In: C.M. Reigeluth (ed.), *Instructional Design Theories and Models*, Hillsdale, NJ: Erlbaum