E^{2}ML: A Visual Language for the Design of Instruction

The last decade has brought about a major change in higher education. Course design has developed from a craftsmanship-like process to a structured production, which involves interdisciplinary teams and requires more complex communication skills. This conceptual article introduces E^{2}ML—Educational Environment Modeling Language—a visual language for supporting complex instructional design processes. E^{2}ML can be used for visualizing the intermediate and final results of design, thus providing documentation in a shared language that can enhance team communication, improve design, and contribute to the development of high-quality instruction. The language and its formal features are presented from a conceptual point of view and illustrated by examples. The main results of a first evaluation study are reported, and the exploitation of E^{2}ML in practice as well as its costs and benefits are critically discussed.

Keywords: Instructional Design, visual language, conceptual language, notation system, design communication.

☐ The discussion about education and technologies in the last decade has brought about a major transformation in teaching: the very idea *course* has been broadened to unexplored dimensions (Bates & Poole, 2003), including Web-based activities, videoconference sessions, high-quality digital media presentations, and so forth. This has made the process of designing courses in several cases a more and more challenging and interdisciplinary process (Szabo, 2002), one that is too complex for one person (Bates, 1999). In some respects, teaching and instructional planning are developing from craftsmanship to a large-scale production process (Cantoni & Di Blas, 2002).

This conceptual article acknowledges this new context and presents E^{2}ML—Educational Environment Modeling Language—an original visual

ETR&D, Vol. 54, No. 3, p. 265–293, 2006

tool for the design of education in complex or large projects. In order to explain the relevance of E^2ML, the first section is devoted to the identification of some features and issues concerning the instructional design (ID) process through the analysis of the literature. In the second section, I introduce E^2ML, justifying its structure from a conceptual point of view, providing applicative examples and pointing out the differences with respect to other ID models. In the third section, I speculate about some issues about the application of a language such as E^2ML to real design practice and report some results from a first evaluation study. In the conclusion, I present a summary along with indications for further work.

COMMUNICATION ISSUES IN ID

The successful integration of technologies in educational activities is not a matter of mere will, nor is the mere decision to use technologies enough to guarantee their successful application in the learning process. When it comes to integrating technologies in education, the subject of teaching, intended as the conception, design, development, and delivery of a formative action, involves an interdisciplinary team (Greer, 1991). The profiles in the team depend on the specific context. Generally, a team should involve "any combination of subject experts or faculty, project manager, instructional designer, graphic designer, computer interface designer, desktop editor, Internet specialist, and media producer, depending on the design of the project" (Bates, 1999, p. 70; see also Achtemeier, Morris & Finnegan, 2003). Another example of labor division in ID is reported by Duffin and Gibbons (2001), along with an analysis of its shortcomings. Each of these professionals makes use of a technical language and misunderstanding is a pitfall that can endanger successful development (effectiveness of communication). Moreover, it is necessary to find a trade-off between the savings due to the specialization of each activity in the process, and the costs of communication among the different actors (efficiency of communication). These problems clearly call for the definition of a standard, or lingua franca, among the different profiles involved in ID.

Other issues are at stake too: how the final learning activity can maintain its overall consistency; how to seamlessly merge the contributions of all profiles into one final product. The overall complexity of the design of instruction can be managed by assigning specific tasks to several specialists and by organizing the production process into phases, following a project management approach (Bates, 1999; Greer, 1992). Several models of ID describe the main phases a well-structured project should undergo, summarized by the basic steps of the ADDIE model (analyze, design, develop, implement, and evaluate), which reflects much of the practice as designers see it (Rosenberg, Coscarelli & Hutchinson, 1999). All authors stress the role that communication plays in the organization and management of the development process (cf. Dick, Carey & Carey, 1996; Morrison, Kemp & Ross, 2004; Smith & Ragan 1999). Cox & Osguthorpe (2003) argued that management activities cover about half the time of the instructional designer's activity. The management of the design and development process is based on what Greer (1992) and Bates (1999) called a blueprint—usually a written text in natural language. Is there a way to produce a more standard and synthetic description of the instruction?

Most ID models include the evaluation and revision phases *at the end* of the design process (e.g., Dick et al., 1996), and some of them suggest a constant tryout and revision process (e.g., Greer, 1992). Although a thorough control of quality is necessary, both solutions are costly, because they take place after the production has started, even if in some cases they involve only prototypes. Is it possible to support at least a partial quality check at design time?

Finally, after a course has been developed, usually the only documentation available is the actual learning materials. This raises some issues in the case where a redesign or adaptation process is required for reuse, especially where the original designer is not available. Is it possible to produce a documentation that can guide the reuse and adaptation of the instruction?

EXISTING LANGUAGES

The literature research conducted during the development of E^2ML revealed the lack of a visual design language for ID. Morimoto, Kogure, Kouno, Yokoyama and Miyadera (2003), in their work on notation systems for lesson plans, also reported the lacking of such a design tool.

Waters and Gibbons (2004) reported that instructional designers use idiosyncratic and "personal" design languages and notation systems, but no complete blueprint language exists. These authors recognized that the work of Horn (1974) and Merrill (1983) represents a step toward the development of a language, but their contributions mainly focus on a particular layer of design—namely Horn on content structures, and Merrill on strategy structures. Eckel (1993) has also proposed an instructional language centered on interaction design.

Very recently, some European researchers have begun working on visual tools for ID, borrowing from learning technology standards and software engineering. EduWeaver (Bajnai & Lischka, 2004; Lischka & Karagiannis, 2004) and the person-centered e-learning patterns (Derntl, 2004; Derntl & Motschnig-Pitrik, 2004) are the first results of such trend.

E^2ML LANGUAGE DEFINITION

E^2ML is a visual language for the design of educational environments. Its general approach is visualization, and its main assumption—which is actually a truism—is that being able to *see* the object being designed may improve the design itself by enabling communication and stimulating reflection. The development of such a language heads the call by Gustafson & Branch (1997), who called instructional developers to be toolmakers and not only tool users in order to face the new challenges of a rapidly changing world.

E^2ML is a tool to develop what Greer (1992) and Reigeluth (1983) called a blueprint: a representation of the instruction that all stakeholders, designers, developers, and instructors can see, understand in a similar way and, it is hoped, agree on. E^2ML is, therefore, a representation language and not an ID model in the classic sense. Traditional ID models, in fact, describe the steps of the development process, whereas E^2ML can be used at any of these steps for describing the instruction being designed. As a language, it should be integrated in the process described by other ID models.

According to the categories proposed by Gibbons and Brewer (2005), E^2ML is a design language, that is, a tool that designers use to communicate designs, plans, and intentions to each other and to the users of their artifacts, with a very limited number of basic concepts, and coupled with a visual notation system.

Document Sets

The development of an E^2ML blueprint means modeling the instruction into a set of defined documents that provide a support for the people involved in the design process. The documentation is organized into three document sets:

1. *Goal Definition,* that is, a declaration of the educational goals. This is composed by two documents: (a) the goal statement, and (b) the goal mapping.

2. *Action Diagrams,* that is, the description of the single learning and support activities designed for the instruction.

3. *Overview Diagrams,* that is, two different overviews of the whole design, (a) the dependencies diagram, and (b) the activity flow.

The documents are described in the following subsections in their standard form. As any real design process and any real instructional situation have their own unique features, they can be adapted (simplified or detailed) to the needs of the specific context or design team. They are produced at different moments in the design process, and do not have a tight correspondence with specific phases. Therefore, the order of presentation should not be understood as an indication of method: The elements of E²ML can be implemented flexibly, in a sequence tailored to the needs of each project.

Goal Definition (Document Set 1)

Expressing learning goals means creating a compass for design, and is important for different reasons: selecting what to teach, how to teach it, what to evaluate, and how to make the whole instruction consistent (Anderson & Krathwohl, 2001; Gronlund, 1995; Yelon, 1991).

Starting from Bloom (1956; 1964), several authors have proposed models and classifications of learning goals, such as Gagné, Briggs, and Wager (1992) and Merrill (1983). Dick et al. (1996) described a method for instructional analysis, that is, for the breakdown of goals into smaller units, thus defining subgoals and entry competencies. Generally speaking, the outcomes of the instruction can be divided on two levels: (a) general goals for the instruction, more loosely defined; (b) specific objectives, concerning a delimited topic or skill.

One central issue for the development of relevant and consistent goals is collaboration with subject-matter experts (SMEs). A second issue is creating a common understanding in the whole team about the goals as primary requirements for the project. For this reason, the topic of the first E²ML document set is learning goals. E²ML focuses on the representation of high-level goals, but the same documentation structure could easily be extended for lower-level objectives.

Goal statement. The goal statement is a table collecting all learning goals (eventually subgoals and objectives). Goals are described by the elements represented in Table 1.

E²ML goal modeling relies on requirements engineering. Assigning goals an importance score can help ranking them, thus indicating priorities in the design process (Bolchini, Paolini & Randazzo, 2003). The importance score can be calculated as a 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 who expressed them (some stakeholders may be weightier than others). Indicating target, stakeholders, and relevance and, eventually, ranking goals is important in order to (a) estimate investments, (b) resolve potential design conflicts, and (c) design more sensible assessment tools.

In order to connect goals and instruction more tightly, the approach and assessment descriptors were introduced. This reveals the choice of E²ML: pro-

Table 1 □ Goal statement definition table.

Element	*Description*	*Example*
Tag	A unique identifier (used for reference)	G1
Statement	A text statement of the goal, expressed from the point of view of the learner. Goals can be detailed by subgoals, expressed in the same form. [Indicate if *goal* or *objective*]	Read and understand a special purpose text in economic English in a proficient way for daily work. *Subgoal* Recall specific vocabulary
Target	The learners who should achieve the goal	The whole class *or* Students who also take the ENG3 course
Stakeholder	Who expressed the goal and has interest in its achievement	The Faculty Dean *or* The Instructor
Approach	How the goal is approached within the instruction (general indication of the method or instructional strategy)	Case study *or* Examples and discussion
Assessment	When (and how) you will test that the goal was achieved. [Indicators of achievement]	Exercise on a text during the written exam *or* Writing a summary of an article from *Business Week*—Achieved if language is correct, and gets all important points
Importance	The relative importance value of the goal in the context of the whole course	5 (out of a 5-point scale)

viding a structure for expressing design decisions rather than methods for making them. E^{2}ML therefore does not define an approach for expressing learning goals, but provides a template where they can be recorded as a reference for the project team. In particular, following the E^{2}ML documentation structure brings two advantages: (a) a reminder that some decisions should be made, simply observing that some part of the document is still empty; and (b) a check of consistency, provided by the opportunity to have a synthetic view of the whole instruction.

The example in Table 2 is taken from a course in institutional communication for third year students in a faculty of communication sciences (referred to as INST3 in the following).

Notice that the goals were formulated by different stakeholders, among whom were the students themselves, who had the opportunity to discuss some features of the course in the first classes and to negotiate G7 as a goal. The importance values clearly set the focus of the course on goals G1 and G3.

If the goal statement is produced before other design phases, the indication

Table 2 ☐ Example of goal statement from INST3.

Tag	*Statement*	*Target*	*Stakeholder*	*Approach*	*Assess-ment*	*Imp.*
G1	Name and define concepts of organization theory for institutions (basis for describing institutions)	All students	Instructor	Explain and discuss	Final exam	5
G2	Understand a framework for the categorization of institutions	All students	Instructor	Examples	Final exam	4
G3	Describe, classify, and compare institutions	All students	Instructor, University curriculum committee	Examples	Final exam	5
G4	A. Analyze institutional problems B. Figure out possible solutions to critical issues	All students	Instructor	Case studies, problem solving	Final exam + Case study work	4
G5	Develop interest in institutions and institutional communication (as a possible professional field)	All students	University curriculum committee	Examples of possible jobs or roles	Whole course (personal engagement)	3
G6	Recall relevant examples and best practices of institutions	All students	Instructor	Examples	Finalexam	4
G7	Design multimedia learning materials about institutional communication	All students	Instructor, Students	Guided practice, problem solving	Case study work	3

Note: Imp. = Importance

of the approach and assessment columns can be taken as hypotheses for the following steps. Otherwise, they can be filled later as a record of validated decisions.

Goal mapping. In order to enhance communication, learning goals can be expressed visually by mapping them on a visual grid or representation, such as Merrill's content-performance matrix (1983), the revised Bloom's taxonomy (Anderson & Krathwohl, 2001), or the QUAIL model (Botturi, 2003). According to Anderson and Krathwohl, trying to classify a goal lets all implicit understandings emerge, and is a chance to align the whole team. Using a visual grid makes classification easier, also allowing people outside the education field, such as technical or media staff, to take part in the discussion. From a practical point of view, visual models can be useful as a negotiation device also with external stakeholders.

Figure 1 provides two examples of visualization of the goals expressed in Table 2 with the Quail model and with the content-performance matrix (G5 is not present in the latter visualization because the matrix does not consider attitudinal goals).

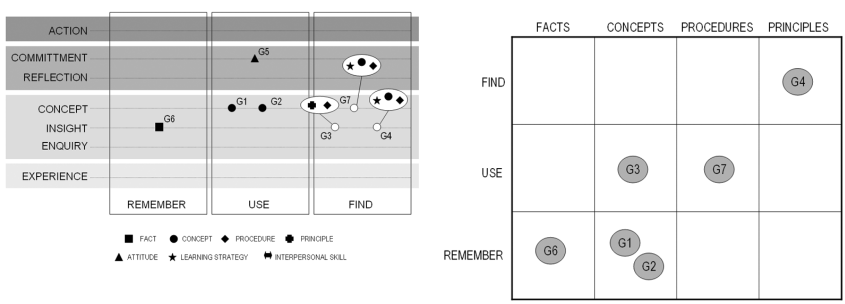
Two points deserve great care: first, the representation device should be consistent with the kind of goals addressed (cognitive, psychomotor, affective, etc.); secondly, the designer should be familiar with the representation and be conscious (if not share) its underlying implications for learning. This is why E²ML simply suggests the use of visuals, leaving the choice to the designer.

Action Diagrams (Document Set 2)

The choice of E²ML is representing an educational environment as a structured system of activities (Szabo, 2002). Interestingly, this is also the choice of IMS Learning Design (IMS, 2003), the only comparable language currently available. IMS clearly states these reasons for its choice:

- A single relatively small vocabulary can be used to express what each instructional approach asks of learners and support staff.
- It allows different pedagogical approaches to be integrated into a single learning design.
- It supports mixed mode delivery, enabling traditional approaches such as face-to-face teaching, the use of books and

Figure 1 ☐ Examples of goal visualization from INST3 with the QUAIL model (left) and the content-performance matrix (right).



journals, lab work, online learning, and field trips, also to be specified as learning activities.

- It enhances pedagogical diversity and creativity.

Another reason for this choice is that, as Gagné, Briggs and Wager (1992) suggested, each instructional module should be considered as an event, as the happening of a set of interactions between the students, the teaching staff and the content.

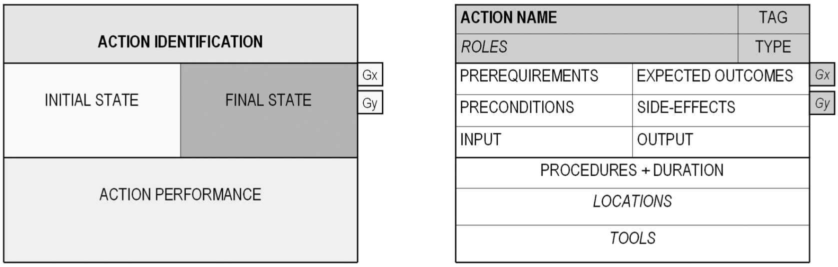
For E^2ML the *action* is therefore the minimal unit of the educational environment. An action is the performance of a set of acts with a unity of purpose by defined acting subjects. Unity of purpose means that the action is aiming at one thing, for example, producing a report, completing an exercise, or achieving the understanding of a concept. The acting subjects could be a single learner along with the tutor, a whole class with the instructor, or a tutor alone. An action can be split in several subactions according to the time and/or space unity criterion in the specific setting (a single lecture or a videoconference). This second distinction (time-space) should, of course, match with the previous one (goal-subject). These considerations should be taken into account for selecting the granularity of the E^2ML representation, which directly depends on the actions defined.

Action structure and types. The general schema for the representation of an action is presented in the left-hand side of Figure 2. The upper part of the diagram contains the proper identification for the action, that is, its identifier tag, name, type and the involved roles (the acting subject). The middle-left area describes the initial state, that is, the necessary and sufficient conditions for learning to be achieved, or for the performance to be successfully completed. 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 action performance, including locations and tools. The squares hanging on the right-hand side are references to the learning goals as defined in the goal statement, thus providing a tight connection between goals and activities. The complete E^2ML action diagram is presented in the right-hand side of Figure 2. The definitions of the elements are provided in Table 3.

Learning actions are directly concerned with the learners' progress with the instruction, and include lectures, discussions, exercises, and personal study. Support actions concern the staff's work for the instruction, such as correcting and evaluating the submissions, setting up materials, and solving logistical issues. Clearly, support actions may have no reference to the learning goals, and a minimal definition of the initial and final states.

The definition of the initial and final state may appear somewhat complicated, but it was conceived in order to be as flexible as possible and, at the

Figure 2 □ Action diagram schema: simple (left) and detailed (right).



same time, fine grained. The three rows crossing the initial state and the final state areas describe different types of conditions:

1. *Prerequisites* and *expected outcomes* describe the learner's starting and arrival points for knowledge and learning in relation with the course goals. Prerequisites are essential and necessary conditions for the learner to achieve learning.

2. *Preconditions* and *side effects* describe the learner's starting and arrival points for knowledge that is not related to the instruction. Fluent English may be a precondition for entering a course in game theory, and a side effect of that course would be improved English fluency—yet language skills are not directly bound to game theory as such. Preconditions are accessory yet necessary conditions for learners to take part in the instruction.

3. *Input* and *output* describe the learner's starting and arrival points for objects used and produced. A typical input may be text and a typical output its summary, done by the students. Although output descriptors catch all that is produced within a course (a project, artifacts, texts, presentations, etc.), inputs may partially overlap with tools. Nevertheless, they are necessary for at least two reasons: (a) to model simple objects that are not listed with the tools because they do not require any design and development effort (e.g., simple text copies); (b) to model output products that are used as input in the following actions (e.g., a paper project used afterwards for developing a Web site).

The performance description can be done in different ways according to the

Table 3 ☐ Action diagram definition table.

Element	*Description*	*Example*
Tag	A unique identifier (used for reference)	PS4
Name	The action name	Practice session 4
Type	The type of action (learning or support)	Learning
Roles	The roles involved in the action, eventually the names of the people	Students (all), Tutor (Mark)
Prerequisites	Competencies or prior knowledge that should have been acquired within the learning environment in order to successfully perform the action	Assessing the difference between internal and external corporate communication; naming at least three types of messages for internal and external communication.
Preconditions	Competencies or prior knowledge necessary to the action but that do not belong to the goals	Proficient use of MS Word (or other text editor)
Input	Object or materials that are provided for the action, but not listed as tools	Acme Web site www.acme.com
Expected Outcomes	The expected learning outcome related to the learning goals, in terms of goal or subgoal achieved	Understand, analyse and evaluate efficiency and viability a corporate communication policy statement
Side-Effects	Competencies or knowledge acquired through the action but not directly addressed by the action (or course) goals	Advanced skills with MS Word
Output	The material product or object designed, developed or realized during the action	A report of 20 pages submitted via email.
Procedure	A description of how the action is to be performed, including specific tasks for each role	Present the case study ("ACME corporate policy") with the related question sheet. Divide students in groups of three and . . .
Location	The location(s) in which the action takes place or *anywhere*	Classroom A34
Tools	The tools and materials exploited in the action	ACME case study, ACME question sheet
Goal Reference	The goals toward which the action moves the learners, although perhaps not achieving them completely *or* use predefined values: PREQ for *prerequisite*, TRAN for *transfer*, EVAL for *evaluation*, INFO for *course information*, MOT for *motivation*	G1, G7, INFO

necessary degree of detail, for example, by describing the single events or acts, by stating tasks for each role involved, or by a more formal description such as a Landamatics algorithm or heuristic method (Landa, 1974, 1993). A simple checklist for procedure description is the following, taken from Dick et al. (1996): goal presentation, motivation management, content presentation (what content, with what tools and media), examples, feedback, grouping criteria, test and assessment, and transfer and retention. Other schemas could be defined for different instructional strategies, such as problem solving and case studies. The procedure descriptor contains a natural language description, so that any guidelines for specifying procedures may be implemented.

The reference to goals is achieved via the identifier tag specified in the goal statement. These are outside the main action box, and do not coincide with the *expected outcomes*. Indeed, the expected outcome for an action could be a subgoal, or even simple knowledge about the course. The idea is that an action often does not *achieve* a goal, rather simply pushes or helps learners toward it. Goals are up to the learners; actions simply provide a method. Moreover, some predefined goals can be indicated:

1. *Preconditions:* The action deals with specific instruction preconditions (e.g., a brush-up technical English workshop for the Game Theory course).

2. *Transfer:* The action is aimed at enhancing the transfer of knowledge from the learning context to the real performance context (e.g., introducing a new procedure in the professional environment).

3. *Evaluation:* The action serves as evaluation (e.g., the final exam, or a project discussion).

4. *Information:* The action is conceived for providing information about the instruction itself.

5. *Motivation:* The action has a specific motivational goal, which can be described with the ARCS model (attention, relevance, confidence, satisfaction) (Keller, 1984, 1987).

Within the instruction, some actions may be optional and others compulsory. A compulsory action should be declared as such in the course syllabus, and it is expected (or it is sensible to expect) that all learners will perform it. Optional actions are included in the design, but learners are free to perform it (e.g., optional readings), or it is unlikely that they all do that, for example, the instructor may suggest taking some time to revise notes every week, but not all students will actually do that. Optional actions are represented exactly like other actions, except that the external box border is dotted.

The examples presented below refer to the course in institutional communication from which goals were also taken. Figure 3 represents two discussions, Discussion 1 and Discussion 2, aiming respectively at G1 and G4, as defined in the goal statement. Figure 4 describes a huge activity, covering 40 hours, about the reformulation of a multimedia case study. The idea of this activity is to take a set of multimedia materials describing a single institution and to define a structure to make them usable and understandable to first year students in communication sciences. Given the size of the activity, it is decomposed into eight smaller subactions, which are described in other activity diagrams (not reported here).

Once more, E²ML supports the description of the instruction, and does not provide a design method. By doing this, it offers tools for not overlooking details, such as the locations or tools required for some activity, and for seeing the project in a structured and synthetic way.

Overview Diagrams (Document Set 3)

The last E²ML document set contains diagrams that provide the big picture, a synthetic view of the whole instruction. Overview diagrams can be used as

Figure 3 ☐ Example of action diagrams for discussion from INST3.

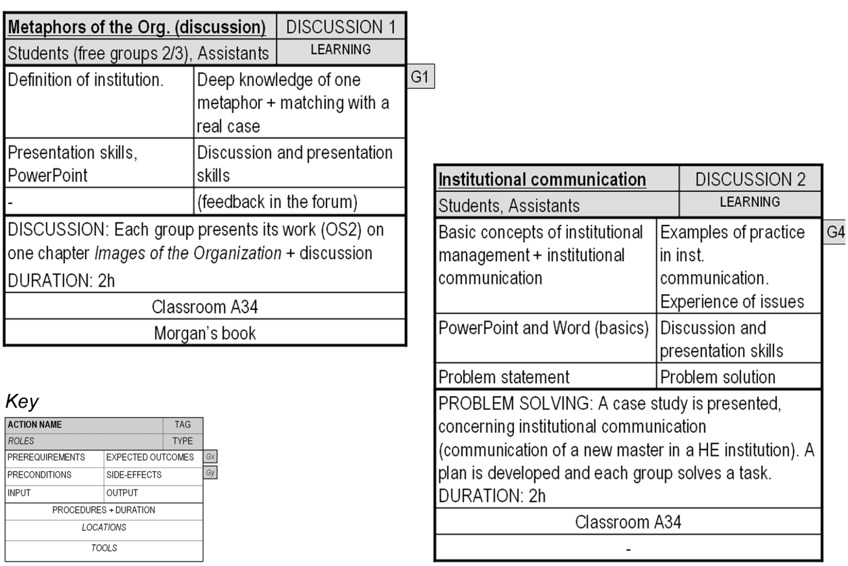
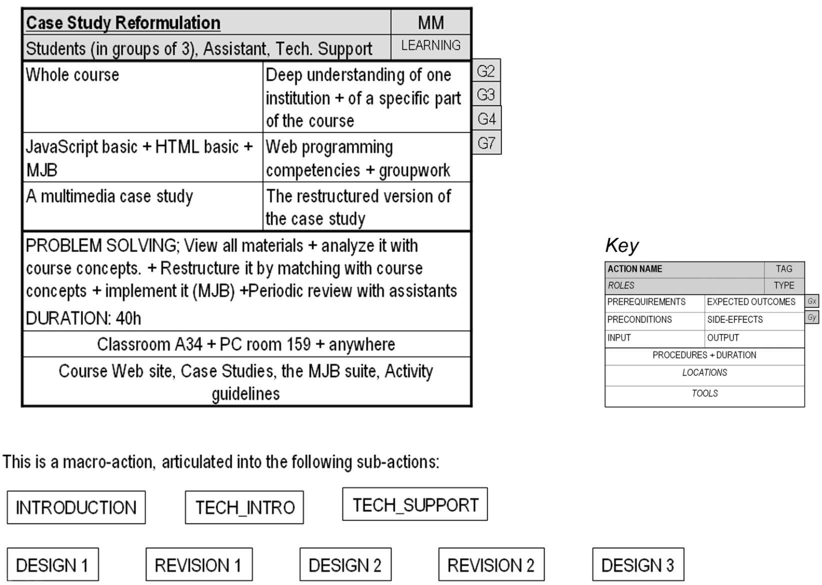


Figure 4 □ The action diagram for the multimedia activity from INST3.



reference for planning the development process and as organizers of other documents and project deliverables. In order to improve legibility, in the overview diagrams actions are represented with a simple box containing their identifier tag.

The dependencies diagram. The idea for the dependencies diagram is that each educational environment has a deep structure connecting its activities in a meaningful way, and that this is not necessarily mimicked in the streamlined disposition of the activities on the calendar. The result is analogous to project evaluation and review technique (PERT) diagrams as they are commonly used in project management. It is useful in order to control the effects that local changes or problems have on the whole environment. The represented relationships are:

1. *Learning prerequisite:* The first action provides a learning outcome that is the prerequisite for the second action (e.g., a lecture provides concepts for the following analysis work).

2. *Product:* The first action produces some artifact that is required as input for the second action (e.g., a group-work activity produces a presentation which is shown during the following class

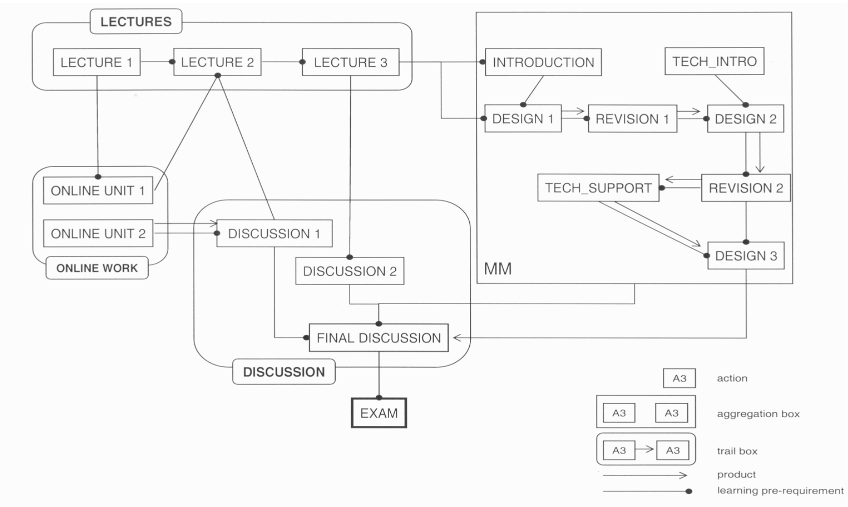
discussion). Product arrows may be tagged with an indicator of the product (e.g., report).

3. *Aggregation:* An activity is part of another activity (it is a subactivity).

For improving legibility, actions can be grouped into *trails*, or logical groups of actions, for example, all lectures, or all the actions that form a specific activity in a course. The key is summarized in Figure 5, along with an example that provides a complete description of the course in institutional communication. It is composed of: (a) three lectures, (b) two online units, (c) three discussion events (one is a final discussion), (d) the multimedia development activity (MM), which includes several subactivities of design revision and technical support, and (e) the final exam.

The dependency diagram groups actions in trails according to their nature (lectures, online activities, discussion and the final group work). The prerequisite relationships reveal the intertwined structure of lectures, activities, and discussions. In order to understand the added value that it brings to design, notice the dependency between Discussion 1 and Lecture 2. A discussion is by nature an open-ended event, as the interaction between the discussants (instructor, assistants, and students in this case) largely determines the results. The fact that Lecture 2 depends on the outcome of Discussion 2, therefore,

Figure 5 ☐ Example of dependencies diagram from INST3 (left) and key (bottom right).



means that the lecturer should be aware of the results of the discussion and be ready to adapt the lecture—commenting on the discussion and, perhaps, filling some gap.

Notice that the dependencies diagram does *not* tell anything about the temporal sequence that actions have in the course. Clearly, if actions are well sequenced, prerequisites and product relationships will be respected.

The activity flow. The activity flow is a visualization of the instruction calendar and provides an overview of the flow of educational activities during the course time span. It is similar to a flowchart diagram that represents each learner's path through the instruction. Actions are therefore sequenced, and if necessary ordered into more parallel branches. Each action can take place at a defined moment in time (e.g., on a particular date or time) or be allocated for free execution within a defined timeframe. In order to make it more useful, the instruction can be divided into phases that are reported in the activity flow, such as course introduction, classes, final reporting, or rehearsal. An even greater level of detail can be reached by representing the action flow on an adequate time grid (e.g., days, weeks, or months) on which all action instances are represented.

Splits (branches) can be added to the action flow as advanced elements:

1. *Conditions*, that is, splits based on conditions of the if-then-else type (e.g., "if the learner's average mark is more than 7.5 out of 10 do activity A, else do activity B").

2. *Options*, that is, unconditional splits in which learners may choose one out of a number of actions.

3. *Selections,* that is, multiple splits (or N-out-of-M splits) that represent the learner's ability to select a certain number of activities out of a given set (e.g., at least two, maximum four activities out of the six proposed).

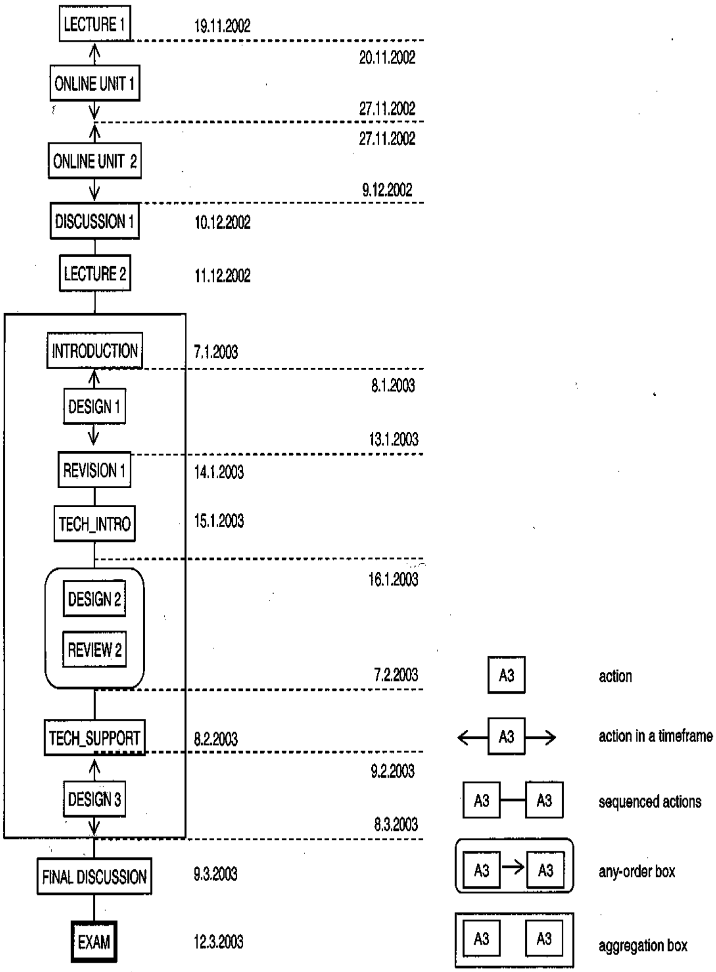
4. *Parallel activities*, that is, splits where all branches have to be completed or all actions executed.

5. *Any-order actions*, that is, branches in the activity path where a number of activities should be completed in any order.

The diagram representation of all splits is an ellipse containing a string identifying the type of split: IF for conditions, OPTION for options, AT LEAST *X* [MAX *Y*] for selections (where *X* is the minimum number of actions to be taken and *Y* the maximum) and ALL for parallel activities. Annotations can be added on the outgoing branches of each split for clarifying their meaning, such as conditional statements. An exception is made for any-order actions, which are represented within a rounded-corner box. Joins are also necessary for a sound notation of splits. For simplicity's sake, E^2ML defines only a generic *join* element, represented by a filled circle. The representation key for the activity flow is summarized in Figure 6.

Obviously, the correct basis for a consistent design of the action flow is the dependencies diagram: Prerequisites and product relationships should be

Figure 6 □ Activity flow example from INST3 (left) and key (right).



maintained in scheduling the activities. Although this may sound trivial, a good calendar might be a challenge when forced by agenda constraints, classroom reservations, or the invitation of external speakers.

The example of activity flow in Figure 6 shows the streamlined nature of the course in institutional communication. The greatest part of actions is to be completed within periods, leaving the students a sort of local flexibility yet allowing synchronization moments in lectures and discussions. Notice that line dates are used for indicating timeframes in which students had to complete activities, and single dates for class sessions.

Sources

Before discussing the integration of E^{2}ML into the design process, it is worthwhile to identify its inspiration sources. E^{2}ML is a specialized process design language, tailored to the needs of education. The activity flow and the dependencies diagram reprise, respectively, flowcharts and PERT diagrams. In particular, the activity flow presents several devices that allow the representation of parallel activities, choices, conditions. E^{2}ML introduces them with original names, more related to the specific instructional context, but they are actually comparable with more general definitions (e.g., Kiepuszewsky, Hofstede, & van der Aalst, 2002; Pattern, 2003).

Other obvious references are modeling languages, such as unified modeling language (UML, 2001) or W2000 (Garzotto, Paolini, & Schwabe, 1993). The visualization of design objects is the key for such models and is explicitly considered here. Other relevant yet scientifically unreferenced sources are architectural blueprints for buildings, with plans, technical details, and 3-D visualizations, and so forth. Waters and Gibbons (2004) pointed out that almost all technical and design disciplines have developed one or more design languages and notation systems.

A third less evident source is requirement engineering (Satcliffe, 2002; see also Sommerville & Sawyer, 1997). According to Bolchini and Paolini (2002), requirements engineering models have three intertwined tasks: (a) supporting the elicitation of requirements; (b) supporting analysis and modeling during design; and (c) supporting the negotiation and validation of design. All of these are goals that E^{2}ML is trying to achieve, too. The modeling of instruction is in fact an implicit requirements specification for the tools developed to support it in terms of scenarios (the actions in which a tool is used are an exhaustive list of possible scenarios for that tool) and goals. Moreover, the goal definition document set is an education-specific answer to the call for lightweight methods for requirement engineering by Bolchini and Paolini. From this point of view, the particular features that E^{2}ML reprised are the definition of goals as external to the actions, the introduction of stakeholders, and goal ranking.

A last source, which influenced the semantic definition of action diagrams, is artificial intelligence, from which the general notion of action as initial state, final state, goal, actors, and resources was taken.

USING E^{2}ML FOR DESIGN

The Benefits of a Design Language

Gibbons and Brewer (2005) analyzed the application and impact of design languages in different fields, such as music, architecture, software engineering, and dance, and came to the conclusion that "Instructional Design will also benefit from this trend as designers and theorists become conscious of the existence and application of design languages and their related notation systems." Their optimistic view is based on the observation that the use of a design language and the related notation system brings huge benefits to the design practice, in particular: (a) It is a tool for remembering designs, thus fostering the creation of a culture based on good practice; (b) it provides a structured problem-solving workspace in which designs can be developed and shared, thus fostering communication and the creation of a community; and (c) it is a kind of laboratory tool for sharpening and multiplying abstract design language categories, that is, it provides a way to express and assess design decisions.

The E^{2}ML notation system, coupled with its design concepts, is proposed here as the basis for the development of a design practice that achieves such benefits in order to improve the quality of instruction.

A First Evaluation

A first evaluation of the perception of usefulness of E^{2}ML was reported in Botturi (2005), and its results clearly indicated that instructional designers see visual languages as a potentially useful tool for their practice. This small study involved a sample of 12 designers employed as course designers or course developers at universities in Canada and the United States, and was focused on the very first introduction of the language in the design practice. Data were collected through focus groups and interviews after a presentation of the language and the discussion of some cases.

The overall impression that all designers expressed during the focus groups is that E^{2}ML looks potentially powerful, flexible, and adaptable to different strategies and situations. They confirmed that they develop a mental image of the course, and that if it can be visualized with E^{2}ML, it can provide "an interesting focus for the discussion." According to their perception, E^2ML is mostly useful for keeping the overall consistency of a course, and in particular for discussing the consistency of goals and instructional activities with the instructors or course authors, as "they usually discuss the goals and then forget them in the actual planning." Designers also think that E^2ML is useful for blueprinting a course, because it "works well in organizing people's thinking," and "may speed up collaboration," also allowing a greater detail than textual blueprints. Finally, it helps to "make the evaluation more evident," identifying activities in which the achievement of specific goals is assessed.

Interviews were conducted to determine if E^2ML actually solved the issues presented in the first section of this article (interdisciplinary communication, requirement analysis, revisions, reuse and adaptation, and project management). All designers agreed that the language provides means to tackle such issues, and were very confident that it can enhance team communication, support the comparison of different designs, and help maintain the overall consistency of the instruction. The use of E^2ML for checking the implementation status received a middle confidence score, while lower confidence was expressed regarding the use of E^2ML for the development of instructional materials, because designers think that it is too high level for implementation. In this respect, it should be noticed that each specific medium has its own design languages that can be used for development anytime the complexity of the project makes it necessary: storyboards, hypermedia design diagrams, maps.

Starting from these results and from the experience made so far with the language, in the following paragraphs I propose some guidelines for the use of E^2ML, expressed as answers to questions, and followed by some indications about the cost effectiveness of the language. The parameters for a more complete future evaluation are reported in the Conclusion and Outlook.

Who Uses E^2ML?

E^2ML was developed for instructional designers, and every effort was made to make it usable, understandable, and practical to them. In the same way they develop their own jargon—specifying such terms as *template* or *blueprint,* or creating such expressions as *round disclosure*—designers should also feel free to take E^2ML, or any of its parts, and extend it, adapt it, and make it suitable to their problems. E^2ML can also be used partially, without exploiting all its features, or using them for only some activities.

Novice designers could use E^2ML as a language for practicing design. From this perspective, having a language means having the ability to focus on design itself, without slipping away to development—which easily happens when learning materials are the only tangible product of the whole process.

Should or could E^2ML visualizations be used with students? E^2ML diagrams are not conceived for them, in the same way that technical blueprints for a two-floor house are not the best support for letting the senior couple that bought it dream about their retirement. Nevertheless, a visualization of the flow of specific activities has been proved to enhance student performance in particular settings, such as problem-based learning (Santoro, Borges, & Santos, 2003). Diagrams could also be used for negotiating some steps in the instruction, and for improving the critical comprehension of the learning process. In order to make them more effective, the style of diagrams should be rearranged and made more appealing.

Where to Start?

E^2ML does not have a unique access point. It is a language, and as such it can be used with different strategies, for example, taken from other ID models. Therefore, goals should be defined first, but the designer could decide to proceed bottom up (resources, actions, overview diagrams) or top down (overview diagrams, actions, resources). It is likely that the documentation would be produced in cycles of refinement (Morrison, Kemp & Ross, 2003). For example, particular instructional patterns or strategies could be selected before starting the design, and goals could afterward be matched to them, or external constraints might force the design process outside the designer's intention. Content is also an issue: Any instructor or SME is likely to feel much more comfortable if beginning with a sort of table of contents, or an outline of *what* will be presented in the course. In short, despite formal models and theories, the design activity as such does not have a unique starting point. This reflects the very nature of human ideas and of human interaction. The flexibility of E^2ML is an attempt to foster—or at least not to hinder—creative unpredictability and serendipity.

Does E^2ML Consider Instructional Strategies and Constructivism?

Some may notice that there is no single place in E^2ML where the instructional strategy is overtly defined. E^2ML does not aim at that, yet it is flexible enough to represent a great variety of different strategies. Moreover, the definition of an instructional strategy is a necessary element for an effective use of any ID model or language. The work of Smith and Ragan (1999) offered insightful descriptions of instructional strategies. Recently, Merrill (2002) presented major guidelines, or first principles, for instructional strategies. Other authors have advocated the design of more open learning environments, following a constructivist approach (Jonassen, 1999; Mayer, 1999), where learners are actively and collaboratively engaged in the construction of their knowledge.

At the present time, no specific contraindication was found against the use of E^2ML for constructivist learning environments.

Emphasizing the distinction between design language (the concepts and rules to combine them) and notation system (the tool to externalize a design) drawn by Waters and Gibbons (2004), E^2ML could also be considered as a basic notation system that could be specialized in order to represent designs developed with specific instructional strategies, rooted in different learning theories.

Is E^2ML All I Need?

E^2ML is not a complete tool in itself, but it becomes a useful means of expression if framed within a structured design process as proposed in traditional ID models, such as the ones quoted above. Gibbons (2003; see also, Gibbons, Nelson, & Richards, 2000) provided an interesting and comprehensive model of the *layers* of ID—E^2ML clearly covers only a part of them, namely the strategy or event, control, and management layers. This calls for future extensions of E^2ML, or for its integration with other notation systems or design languages.

Moreover, while E^2ML provides a sound representation of the instruction, it does not cover all the documentation to be produced in a single project. Other complementary languages could and in many instances must be used, such as Web design models for the development of Web resources. Additionally, the document sets do not collect all the possible documents produced by designers, because each designer or team has its own way of proceeding.

How Much Effort on Design?

A comparison provides a possible answer to this question, which is more a matter of one's idea of teaching and learning than a technical subject. Teaching is an art, and it cannot be completely and definitely designed, like a program that is then executed and from which no surprise is expected. In the design of education, a small detail may have a great effect in the long run: Although designed in a short time, an instructional unit may require several days of effort from the learners, thus being a relevant change factor for them (Schwier, Campbell, & Kenny, 2003).

The design of instruction could be compared to the *canovaccio* (the plot) in the 18th-century Italian *comedia dell'arte.* In this kind of comedy, the actors played the roles of traditional masks in the Carnival of Venice and in other popular celebrations, and they did not have a script of their part. They simply knew the general plot, as follows:

> The servant plans to play a joke on his master, but the mistress

> discovers it. However, as she has a lover, she does not reveal it to her husband, but, disguised, offers support to the servant. The joke succeeds, but the mistress falls in love with the servant and flees with him, leaving her former lover under police suspicion of being the real author of the joke.

Once on the stage, the actors acted in a way consistent with the plot, but their actions were hectic, reacting to what the other actors said, and introducing new elements—and this was their art. The designer's and the instructor's art is much the same: A plan or E^{2}ML blueprint can give shape to the interactions with the learners, but improvisation and reaction to the unexpected is the rule—given that the final outcome, or one even richer, is reached.

Is It Worth the Candle?

How cost effective is E^{2}ML? A first necessary consideration concerns its costs. As any technical language, E^{2}ML must be learned, and designers need some time in order to get acquainted with it. Experience up to now has shown that a two-day workshop is sufficient to make expert designers fluent in its use for their professional practice, whereas about twice that time is required to train novice designers. Fluency in the use of E^{2}ML means the ability to read the diagrams, to compare them, and to sketch new ones with paper and pencil (i.e., without using a specific software application or any graphic design tool) both as personal "think-tool" and as communication device. According to research in visuals for design (a good summary is provided by Blackwell, 1997), it is likely that experienced designers will use the diagrams in different ways and with more details than do novices.

Using E^{2}ML for the design of a single course requires a relatively small overhead for the creation of diagrams, depending on the extent to which the language is used and the familiarity of designers with it. For example, using E^{2}ML as a personal think-tool with pencil and paper requires almost no overhead. This is a current practice in some Swiss Virtual Campus (SVC, 2005) e-learning projects developed at my institution. There, instructional designers are involved in large collaborative projects with faculty members from different Swiss universities, and they often incur communication problems due to the different languages being used (Italian, German, French, and English) and to the different academic traditions. During meetings, instructional designers create diagrams as a visual form of taking notes, and use them mainly for discussing learning scenarios with SMEs and other stakeholders. They then take some time (about an hour) after each meeting for refining them. For some projects, E^{2}ML is also being used as a standard language to archive designs. This practice requires, for each learning unit, one or two hours of extra work to create digitized versions of the designs.

The overhead is usually spanned through the different phases of the process, because the diagrams are refined at different times. The cost for first designs can be reduced by developing diagram templates or by using a specific software application, and by developing diagrams that are suited only to the phase the design process is currently in. There is no need for graphically full-fledged digital diagrams for a brainstorming session.

According to some designers, such costs are balanced by better and faster focusing in meetings and with nondesigners (Botturi, 2004). Subsequent redesigns or adaptations of the course can start on a more solid basis if E^2ML is used, because the designer can take advantage of the visual description of the instruction and not only of the learning materials. From a general point of view, a complete E^2ML documentation is a detailed plan of the instruction, and is a valuable tool for project management (cf. McGriff, 2000).

This article presents E^2ML *Core Version*, however, an *Advanced Version* has been developed as well, which is more formal and more expressive, but also more time-consuming to use. This version provides a formal definition of all the elements in the instruction, namely *roles and actors*, *tools and materials*, and *locations*, along with the ability to express abstract reusable actions. E^2ML Advanced Version complies with the IMS Learning Design standard (IMS, 2003). This means that the description of a unit of instruction could be automatically translated into the corresponding extensible markup language (XML) description. In this sense E^2ML could be used as a visual interface for making XML and metadata standards usable for designers and educators.

CONCLUSIONS AND OUTLOOKS

The challenges of the last decades in higher education include a shift in the idea of course, which results in a more complex and challenging environment for instructional designers. This is particularly true from the point of view of communication: A design team is by nature interdisciplinary and in interaction with stakeholders, external partners, and other design teams. E^2ML was conceived as a visual language that can smooth and enhance project communication. In this article, the structure of the language, articulated in three document sets, was presented, along with some examples. The main idea is modeling the instruction as a set of interrelated actions, aimed at goals and performed by actors with specific roles exploiting tools and locations. Some issues concerning its integration in the design process were discussed.

The most natural step after the development of such a language is a thorough evaluation of its impact on the design practice. The evaluation of a language is no easy task, because the use of languages is the result of complex interactions among the speakers, and among the community of speakers and other communities, and their effectiveness is tightly connected to creativity. In some sense, a language is a flexible and continuously developing tool. The complex connection between these elements makes it difficult to define an evaluation protocol. Although some courses would benefit from it (for example a mixed-mode course), other courses even in the same institution might not (for example a face-to-face lecture series). At the same time, some designers may feel comfortable enough to use it for the quick design of small courses, where it might otherwise not be useful.

The quality of a tool is its adequacy to a problem-solving activity for its users. Given the complexity of ID, specific and limited subactivities could be observed, and this may provide elements for evaluation. An example would be a new designer in charge of redesigning two courses developed by someone else: The designer has only the course materials for the former, and a complete E^2ML documentation for the latter. An evaluation of the work, and of the aid of the documentation, along with a measure of effectiveness (e.g., time spent), would offer a measure of the impact of E^2ML on a particular situation. Other small scenarios could be identified, such as the redesign of a unit, or the adaptation of a course to a different target. In the same way, specific communication events could be observed as part of the subactivities of the design process. The use of diagrams for involving stakeholders could be another interesting point.

The assessment of the institutional and organizational changes that E^2ML would bring to a community of designers would provide additional relevant elements. The impact of a language, in fact, should be also observed on the social dimension. The ability to create a shared repository of courses, or to define pedagogical patterns (cf. Belfer & Botturi, 2003; 2004), is likely to change interactions among designers. An evaluation would include the training and integration process of novice designers, the sharing of expertise and best practices, the reuse of design, and communication inside and outside the team—as elements of knowledge management. The guidance of the transformation would also be at stake: Who is sponsoring the exploitation of E^2ML? What are the major drivers? What are the perceived benefits and fears? Diffusion theory (cf. Rogers & Shoemaker, 1971) would provide a solid background for this part of the evaluation.

A collection of case studies would contribute to the improvement of E^2ML in two respects: (a) by providing a set of test cases about its expressive power, indicating which features would be a sensible extension and which are pointless; and (b) by providing a repository of cases from which to extract pedagogical patterns, which could be profitably expressed with E^2ML (Belfer & Botturi, 2003). Such a collection would also provide data about the limitations of the language, and about the situations in which it is sensible to apply it.

Finally, a complete integration of E^2ML in design practice could be fostered by the availability of a software application supporting the creation of visuals and the management of the documentation.

The main assumption behind E^2ML is that a visual language may enhance communication, and enhanced communication may improve design; improved design may increase the quality of educational programs. The proposal of this language also contains a call to instructional designers to create tools for communicating their job to nondesigners and stakeholders. In the context of teaching and learning, E^2ML is a sort of shovel, which the experienced gardener can use in a number of ways, learning from his or her own experience, to let the seeds grow into trees. □

Luca Botturi [luca.botturi@lu.unisi.ch] is with the NewMinE Lab and the e-Learning Applications Lab (eLab) at the University of Lugano, Switzerland.

ACKNOWLEDGMENTS

The author wishes to thank Rick Kenny, Athabasca University, for his generous support in this research and for his advice and comments. The feedback of *ETR&D* reviewers during the review process largely contributed to the improvement of the article and to making clear the ideas it conveys.

REFERENCES

Achtemeier, S. D., Morris, L. V., & Finnegan, C. L. (2003). Considerations for developing evaluations of online courses. *Journal of Asynchronous Learning Networks*, *7*(1). Retrieved January 27th, 2005, from http://www.aln.org/publications/jaln/v7n1/v7n1_achtemeier.asp.

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching and assessing. A revision of Bloom's taxonomy of educational objectives*. New York: Addison Wesley Longman.

Bajnai, J., & Lischka, J. (2004). Planning and simulation in an e-Learning engineering framework. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA, Lugano, Switzerland, 2004*, 3265–3269.

Bates, T. W. (1999). *Managing technological change*. San Francisco: Jossey-Bass.

Bates, T. W., & Poole, G. (2003). *Effective teaching with technologies in higher education*. San Francisco: Jossey-Bass.

Belfer, K., & Botturi, L. (2003). Pedagogical patterns for online learning. *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education ELEARN, Phoenix, Arizona, 2003,* 881–884.

Belfer, K., & Botturi, L. (2004). *Online learning design with pedagogical patterns.* Paper presented at the SALT Orlando Conference 2004, Orlando, Florida, USA.

Blackwell, A. F. (1997). Correction: A picture is worth 84.1 words. *Proceedings of the ESP Student Workshop, Washington, DC, 1997,* 15–22.

Bloom B. S. (1956). *Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. Handbook I: Cognitive domain.* New York: Longmans, Green & Co.

Bloom B. S. (1964). *Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. Handbook II: Affective domain.* New York: Longmans, Green & Co.

Bolchini, D., & Paolini, P. (2002, November). *Capturing web application requirements through goal-oriented analysis*. Paper presented at the 5th Workshop on Requirements Engineering, Valencia, Spain.

Bolchini, D., Paolini, P., & Randazzo, G. (2003). Adding hypermedia requirements to goal-driven analysis. *Proceedings of the 11th IEEE International Conference on Requirements Engineering RE, Monterey, California, 2003,* 127–137.

Botturi, L. (2003). E^2ML–Educational environments modeling language. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA, Honolulu, Hawaii, USA, 2003,* 304–311.

Botturi, L. (2004). Visual languages for instructional design: An evaluation of E^2ML. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA, Lugano, Switzerland, 2004,* 234–250.

Botturi, L. (2005). Visual languages for instructional design: An evaluation of the perception of E^2ML. *Journal of Interactive Learning Research, 16*(4), 329–351.

Cantoni, L., & Di Blas, N. (2002). *Teoria e pratiche della comunicazione* [Theory and practices of communication]. Milano, Italy: Apogeo.

Cox, S., & Osguthorpe, R. T. (2003). How do instructional designers spend their time? *Tech Trends, 47*(3), 45–47.

Derntl, M. (2004). The Person-Centered e-Learning Pattern Repository: Design for Reuse and Extensibility. *World Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA, Lugano, Switzerland, 2004*, 3856–3861.

Derntl, M., & Motschnig-Pitrik, R. (2004, March). *Patterns for blended, person-centered learning: Strategy, concepts, experiences, and evaluation.* Paper presented at the Symposium on Applied Computing SAC, Nicosia, Cyprus.

Dick, W., Carey, L., & Carey, J. O. (1996). *The systematic design of instruction* (4th ed.). New York: Harper Collins College Publishers.

Duffin, J. W., & Gibbons, A. S. (2001). Decompressing and aligning the structures of CBI design. *Proceedings of the International Conference on Advanced Learning Technologies, Madison, Wisconsin, USA, 2001*, 61–64.

Eckel, K. (1993). *Instruction language: Foundations of a strict science of instruction*. Englewood Cliffs, NJ: Educational Technology Publications.

Gagné, R. M., Briggs, R., & Wager, W. (1992). *Principles of Instructional Design* (4th ed.). Fort Worth, TX: Harcourt Brace Jovanovich College Publishers.

Garzotto F., Paolini P., & Schwabe, D. (1993). HDM–A model-based approach to hypertext application design. *ACM Transactions on Information Systems, 11*(1), 1–26.

Gibbons, A. S. (2003). What and how designers design? A theory of design structure. *TechTrends*, *47*(5), 22–27.

Gibbons, A. S., & Brewer, E. K. (2005). Elementary principles of design languages and design notation systems for instructional design. In J. M. Spector, C. Ohrazda, A. van

Schaack, & D. A. Wiley, *Innovations in instructional technology: Essays in honor of M. David Merrill* (pp. 111–130). Mahwah, NJ: Lawrence Erlbaum Associates.

Gibbons, A. S., Nelson, J., & Richards, R. (2000). *The architecture of instructional simulation: A design for tool construction* (Center for Human-System Simulation technical report). Idaho Falls, ID: Idaho National Engineering & Environmental Laboratory.

Greer, M. (1991). Organizing and managing the instructional design process. In L. J. Briggs, K. L. Gustafson & M. H. Tillman (Eds.), *Instructional Design: Principles and applications* (2nd ed., pp. 315–343). Englewood Cliffs, NJ: Educational Technology.

Greer, M. (1992). *ID project management: Tools and techniques for instructional designers and developers*. Englewood Cliffs, NJ: Educational Technology.

Gronlund, N. E. (1995). *How to write and use instructional objectives* (5th ed.). Englewood Cliffs, NJ: Prentice Hall.

Gustafson, K. L., & Branch, R. M. (1997). Revisioning models of instructional development. *Educational Technology Research and Development*, *45*(3), 73–89.

Horn, R. E. (1974). Information mapping. *Training in Business and Industry, 11*(3). Retrieved January 27th, 2005, from http://www.stanford.edu/rhorn/a/topic/stwrtng_infomap/artclInfoMappingTraining.pdf.

IMS. (2003). *IMS Learning Design specification* [information model, XML binding and best practice and implementation guide]. Retrieved January 27th, 2005, from www.imsproject.org.

Jonassen, D. (1999). *Designing constructivist learning environments*. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: An overview of their current status: Vol. 2* (pp. 215–239). Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Keller, J. M. (1984). The use of the ARCS model of motivational design in teacher training. In K. E. Shaw & A. J. Trott (Eds.), *Aspects of educational technology: Vol. XVII. Staff development and carreer upgrading* (pp.140–145). London: Kogan Page.

Keller, J. M. (1987). Development and use of the ARCS model of motivational design. *Journal of Instructional Development*, *10*(3), 2–10.

Kiepuszewsky, B., Hofstede, A. H. M., & van der Aalst, W. M. P. (2002). *Fundamentals of control flow in workflow* (QUT technical report FIT-TR-2002-02). Brisbane, Australia; Queensland University of Technologies.

Landa, L. N. (1974). *Algorithmization in learning and instruction*. Englewood Cliffs, NJ: Educational Technology.

Landa, L. N. (1993). *Landamatics ten years later* [interview with L. Landa]. *Educational Technology*, *33*(6), 7–18.

Lischka, J., & Karagiannis, D. (2004). *Modeling and execution of e-Learning resources*. Paper presented at the Symposium on Applied Computing SAC, Nicosia, Cyprus.

Mayer, R. H. (1999). Designing instruction for constructivist learning. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: An overview of their current status: Vol. 2* (pp. 141–160). Hillsdale, NJ: Lawrence Erlbaum Associates.

McGriff, S. J. (2000). *Project management for Instructional Design in Higher Education*. Unpublished manuscript, Pennsylvania State University, USA.

Merrill, M. D. (1983). Component cisplay theory. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: An overview of their current status: Vo.l I* (pp. 279–333). Hillsdale, NJ: Lawrence Erlbaum Associates.

Merrill, M. D. (2002). First principles of instruction. *Educational Technology Research & Development*, *50*(3), 43–59.

Morimoto, Y., Kogure, T., Kouno, S., Yokoyama, S., Miyadera, Y. (2003). A meta-language for a teaching-plan-making support system. *Proceedings of the Wold Conference* *on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA, Honolulu, Hawaii, USA, 2003*, 531–538.

Morrison, G. R., Kemp, J. E., & Ross, S. M. (2004). *Designing effective instruction* (4th ed.). New York: Wiley & Sons.

Pattern. (2003). *Flow control patterns*. Retrieved January 27th, 2005, from tmitwww.tm.tue.nl/research/patterns.

Reigeluth, C. M. (1983). *Instructional-design theories and models: An overview of their current status*. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Rogers, E., & Shoemaker, E. (1971). *Communication of innovations: A cross-cultural approach* (2nd ed.). New York: The Free Press.

Rosenberg, M. J., Coscarelli, W. C., & Hutchison, C. S. (1999). The origins and evolution of the field. In H. D. Stolovitch & E. J. Keeps (Eds.), *Handbook of human performance technology* (2nd ed., pp. 24–46). San Francisco: Jossey-Bass.

Santoro, F. M., Borges, M. R. S., & Santos, N. (2003). Using workflow concepts to support collaborative project-based learning. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications, Honolulu, Hawaii, USA, 2003*, 140–147.

Satcliffe, A. (2002). *User-centered requirement engineering*. Heidelberg, Germany: Springer.

Schwier, R. A., Campbell, K., & Kenny, R. F. (2003). *Instructional designers as change agents: Perceptions of communities of practice*. Paper presented at the Annual Conference of the Association of Media and Technology in Education in Canada AMTEC, Montreal, Quebec, Canada.

Smith, P. L., & Ragan, T. J. (1999). *Instructional design*. New York: Wiley & Sons.

Sommerville, I., & Sawyer, P. (1997). *Requirement engineering. A good practice guide*. New York: Wiley & Sons.

SVC. (2005). *Swiss Virtual Campus Web site*. Retrieved January 26th, 2005, from www.virtualcampus.ch.

Szabo, M. (2002). Competencies for educators. In H .H. Adelsberger, B. Collis, & J. M. Pawlowsky (Eds.) *Handbook on information technologies for education and training* (pp. 381–397). Berlin, Germany: Springer.

UML. (2001). *UML 1.5 specification*. Retrieved January 27th, 2005, from www.uml.org.

Waters, S., & Gibbons, A. S. (2004). Design languages, notation systems, and instructional technology: A case study. *Educational Technology Research & Development 52*(2), 57–68.

Yelon, S. L. (1991). Writing and using instructional objectives. In L. J. Briggs, K. L. Gustafson, & M. H. Tillman (Eds.), *Instructional design: principles and applications* (2nd ed., pp. 75–121). Englewood Cliffs, NJ: Educational Technology.