Technology-Based Instructional Design: Evolution and Major Trends

53

Gilbert Paquette

Abstract

This chapter surveys ICT-based tools and methods that support instructional designers in planning the delivery of learning systems. This field has evolved since the 1970 through several paradigms: authoring tools, expert systems and intelligent tutoring systems, automated and guided instructional design, knowledge-based design methods, eLearning standards and social/cognitive Web environments. Examples will be given to illustrate each paradigm and the major trends will be uncovered. ICT has evolved rapidly, enabling new approaches to emerge, helping more people to design learning environments and building learning design repositories. More and more people are learning on the Web, using learning portals, information pages and interacting with other people, but still with insufficient educational support. New challenges make this field an exciting and blooming research area that has a bright future.

Keywords

Instructional design • Instructional engineering • Knowledge-based design • Educational modeling • eLearning standards • Web-based learning environments

Introduction: Defining the Field

Some authors trace the origin of *Instructional Design* to John Dewey, who, a century ago, "called for the development of a linking science between learning theory and educational practice" (Reigeluth, 1983, p. 5; Dewey, 1900). Others (Dick, 1987) situate the beginning of ID after World War II. But it is really at the beginning of the 1960, that we see the beginning of the new discipline, mainly under the influence of the work of B. F. Skinner on programmed instruction, Jerome Bruner on the cognitivist approach and David Ausubel (Reigeluth, 1983). In the 1970s and 1980s, research on instructional theories blossomed as illustrated by t he: (a) the development of a cybernetic approach (Landa, 1976), (b) the exposure of learning conditions (Gagné, 1985), (c) the identification of instructional strategies based on structural learning theories (Scandura, 1973), (d) the development of a cognitive teaching theory based on enquiries (Collins & Stevens, 1983), and (e) the analysis of instructional strategy components (Merrill, 1994).

Based on these various research efforts, Instructional design is today a collection of theories and models helping to understand and apply instructional methods that favor learning. Instructional Design as a method or a process helps produce plans and models describing the organization of learning and teaching activities, resources and actors' involvement that compose an instructional system or a learning environment. Compared to the theories developed in educational psychology, instructional design can be seen as a form of engineering aiming to improve educational practice. Its link with educational science is analogous to the link between engineering methods and the physical sciences, or between medicine and life sciences.

G. Paquette (✉)
CICE Research Chair, LICEF Research Center, Télé-université du Québec, 100 Sherbrooke St, West Montreal, QC, Canada
e-mail: gilbert.paquette@licef.ca

J.M. Spector et al. (eds.), *Handbook of Research on Educational Communications and Technology*,
DOI 10.1007/978-1-4614-3185-5_53, © Springer Science+Business Media New York 2014

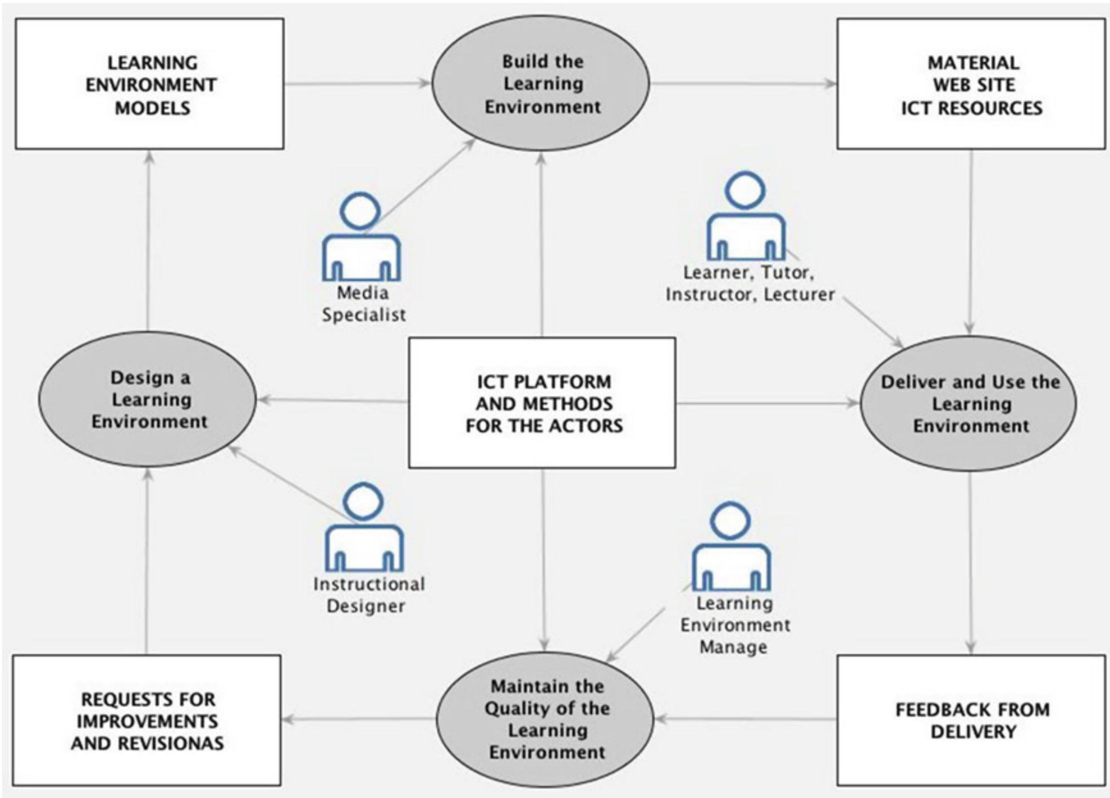

Fig. 53.1 The basic life cycle of a learning environment

The life cycle of a learning environment is presented in Fig. 53.1. This figure shows four main processes going from creation or design, production of a learning environment, and then to its delivery. Finally, a maintenance and revision process serves to detect deficiencies revealed by the delivery of the learning system, leading to improvements proposed to the instructional designers, closing up the loop and starting a new cycle.

Figure 53.1 also shows the products of each process and the main actors that produce them. While there is a sequential progression between these main processes, it is best to picture the global process with subprocesses more or less parallel, sharing information between them with frequent interaction between the actors. In this chapter, we will focus on the instructional design (ID) process, methods, and support tools, but in some case, we will identify the interaction of pure ID with the other three processes, in particular with the production process.

Using this general picture of an instructional system, the following sections will present the main paradigms that propose ways to use information and communication technologies (ICT) to support the instructional design process. These paradigms are authoring tools and languages, knowledge modeling of instructional design methods, automated and guided instructional design, eLearning standards and social/semantic Web environments. Finally, in the last section, we identify the major trends and issues, synthesizing the evolution of Technology-Based Instructional Design.

Authoring Tools and Languages

The use of computers in education started 50 years ago, at the beginning of the 1960s. The first applications were influenced mainly by programmed instruction strategies (Crowder, 1959; Skinner, 1954). Most authoring tools and languages for computer-assisted instruction were limited to present information, ask a question and branch to another unit. Two early authoring systems attempted to go beyond such simple templates, in order to provide more complete learning strategies.

One specialized programming languages, TUTOR, was developed starting in 1965 for use on the PLATO system at the University of Illinois at Urbana-Champaign. TUTOR had powerful answer parsing and answer judging commands, and it had features to simplify student records by instructors. TUTOR's flexibility, in combination with PLATO's computational power (running on what was considered a supercomputer in 1972), also made it suitable for the creation of games and simulations that could be used for learner-centered education.

Later, templates were developed to ease the programming part of courseware creation. For example, (Schultz, 1975) presents MONIFORMS, a set of partially completed coding formats in the TUTOR language that could be adapted by instructional designers in order to implement instructional tactics.

The TICCIT system (Merrill, Schneider, & Fletecher, 1980) attempted to provide built-in complex instructional templates in the mid 1970s. The student had access to a set of learner-controlled keys: Rule, Example, Practice, Objective, Help, Advice, Easy, Hard, and Map. The author provided information accessible behind these keys, to be displayed to the student studying some the rules and concepts for which the information provided. The system also provided a map or hierarchy diagram from which the student could choose the next content to study, but with some help from the system.

With the advent of multimedia and Internet technologies, there has been an explosion of the number of authoring tools. Widely used commercial tools have included Macromedia's Authorware, IconAuthor and Click2Learn's ToolBook. More recent learning content management systems (LCMSs), such as BlackBoard, Learning Space, TopClass, WebCT, and Moodle, are totally oriented towards building Web-based courses. There has been also a proliferation of authoring tools providing templates. However, not many of them offer multiple instructional strategies (Liao, Lo, Oyuki, & Wing Li, 2003).

Moreover, while LCMSs, authoring tools or templates help produce resources for delivery environments based on the more or less limited set of strategies they support, they are essentially helping in the production process. They do not provide much support for instructional designers to analyze learning needs, structure target knowledge and competencies, integrate resources in learning scenarios or plan the production of resource and delivery environment. In particular, they provide no help to select teaching/learning strategies before deciding which authoring tools or templates should be used.

Modeling Instructional Design and Job Aids

With the evolution of technology-based learning, the instructional designer must make a larger set of interrelated decisions. What kind of delivery model shall we use: classroom, Web based, blended? What kind of learning activities do we need for this course? Should it be predefined, offer multiple learning paths or be learner-constructed? Which actors will interact at delivery time, what are their roles, what resources do they need? What kind of interactivity or collaboration should be included? What materials can be reused, adapted or built anew? How distributed resources are to be managed on the networks? What kind of eLearning standards will be used? How can we support interoperability and scalability of the learning system? How can we promote their reusability, sustainability and affordability? To cope with all these decisions and others, an instructional design methodology and a tool set are needed more than ever.

The MISA instructional systems engineering method (Paquette, Aubin, & Crevier, 1994; Paquette, 2004) is a long-term effort to address these new needs of the instructional designers. It has provided a mature methodology at the turn of the century that continues to evolve. As shown in Fig. 53.2, MISA is structured into six phases and four axes under which the main 35 design tasks and their subtasks are distributed. The four axes are deployed from construction of the model or document its properties.

The MISA method is the result of applying knowledge engineering to the instructional design domain. Using the MOT language and editors, the products, the task and the principles of instructional design have been modeled and their interactions identified. The relationship between tasks is represented using a process graph for each of the phases and each of the axes. The design documents produced by each of the 35 main tasks are modeled as concept objects with a certain number of attributes that have well-defined values. The knowledge model describing MISA ensures the consistency of the method. It also help guide the navigation of the designer through the method. Contextual help or intelligent advice can be given by a supervisor or a software agent for each design task, based on the relationships between it and the other tasks in the method and also on the consistency of values for the different attributes in a design document.

The complete model of the MISA method enabled the production of computerized Job aids or design tools. The first one was AGD, a standalone performance support system for ID (Paquette et al., 1994). Later, an improved version of MISA enabled the construction of job aids as a set of Word and Excel templates, supplementing the MOT visual knowledge editor. In 2001, a WEB tool, ADISA was built and is presented in the next section. More recently, MISA/ADISA design scenarios can be edited and processed by the ontology-driven TELOS system (Paquette & Magnan, 2008).

Expert Systems and Automated/Guided ID

Beginning also in the 1990s, expert systems and artificial intelligence techniques started to be applied to the field of instructional design to provide methodological support and intelligent help (Winkels, 1992) to instructional designers. Many expert systems were built for focused ID tasks where they have had generally more success than more general applications (Locatis & Park, 1992). A second category of systems is concerned with helping designers construct Intelligent Tutoring Systems (Wenger, 1987); the Generic

Phase 1- Definition	100 Organization's Training System 106 Present Situation	102 Training Objectives 108 Reference Documents	104 Learners' properties	
	Knowledge Axis	**Pedagogy Axis**	**Media Axis**	**Delivery Axis**
Phase 2 – Initial solution	210 Knowledge Model Principles **212 Knowledge Model** **214 Competencies**	220 Instructional Principles **222 Event Network** 224 Learning Unit Properties	230 Media Principles	240 Delivery Principles 242 Cost-Benefit Analysis
Phase 3 – Architecture	*310 Learning Unit Content*	**320 Learning Scenarios** 322 Activity Properties	330 Development Infrastructure	340 Delivery Planning
Phase 4 – Detailed Design	*410 Learning Resource Content*	420 Learning Resource Properties	430 Learning Resource List **432 Media Models** 434 Media Elements 436 Source Documents	**440 Delivery Models** 442 Actors and resources 444 Tools and Telecom 446 Delivery Services
Phase 5 – Eval	540 Test Planning	542 Revision Decision Log		
Phase 6 – Delivery Plan	610 Knowledge/ Competency Management	620 Actors and Group Management	630 Learning System/ Resource Management	640 Maintenance/ Quality Management

Fig. 53.2 Overview of the MISA instructional system design method

Tutoring Environment (GTE), is a good representative of that category of system (Elen, 1998). We will here focus on a third category of Expert System applications that aim to support the general Instructional Design process. We present here three of them:

- ID Expert (Merrill, 1998), an expert system for designing courseware, which evolved into a commercial system called Electronic Trainer
- GAIDA/GUIDE (Spector, Polson, & Muraida, 1993) provides a guided approach to ID Advising
- Templates and the intervention of an intelligent advisor

The purpose of ID Expert and Electronic Trainer is to provide a consultation system that could be used by inexperienced instructional designers to assist in instructional design decision-making, prior to the programming stage. The expert system gathers information from the user/designer and makes recommendations on the goal of instruction, the content structure that corresponds to the goal, the elaboration of the content structure, the modules that are necessary for teaching the content, the instructional transactions that are best for each module and guidance for elaborating and instantiating each transaction. The output of the consultation is a design specification that provides a skeleton from which instructional materials can be built. The domain of the first ID Expert was limited to goals involving concept classification with a kind-of taxonomies content structure and goals involving procedures for device operation with a path algorithm content structure. ID expert 2.0 extended the initial set of goals and provided a delivery interface. The commercial Electronic Trainer linked the ID expert to authoring capabilities that produced the corresponding learning material. Unlike many expert systems, which are directed toward a single main decision, the ID expert makes recommendations on a series of decision and allows the designer to confirm each recommendation as the reasoning proceeds.

The GAIDA advisory system was developed to support lesson design as part of the Advanced Instructional Design Advisor project at Armstrong Laboratory (Spector et al., 1993). The system uses completely developed sample cases to help less experienced instructional designers construct their lesson plans. GAIDA is designed explicitly around the nine events of instruction (Gagné, 1985). It allows users to view a completely worked example, shown from the learner's point of view (see Fig. 53.3). The user can shift from this learner view to a designer view that provides an elaboration of why specific learner activities were designed as they were.

ADISA is the successor of the AGD system. It is a Web-based system developed to enhance the performance level of instructional designers, in particular to assist teams who

Fig. 53.3 A screen from GAIDA/GUIDE

create Web-based distance learning courses. It embeds a large set of educational knowledge including 17 typologies of educational concepts from the MISA 4.0 method, each offering a set of options for the designer to choose from. It provides an editing part for 35 documentation elements (DE), either forms or graphic models to be produced by tasks of the MISA method. An important feature is the data propagation from one DE form or model to another, based on the MISA 4.0 process models.

What can be learned from the research on automated or semiautomated ID systems? First, productivity improvements have been observed due to performance support

While results vary, using design support tools can achieve an order of magnitude improvement in the productivity of a design team. Second, learning can result for designers using such systems. GAIDA has been evaluated in numerous settings with both novice and expert designers (Gettman, McNelly, & Muraida, 1999). Findings suggest that expert designers found little use for GAIDA, whereas novice designers made extensive use of it for about 6 months and then no longer felt a need to use it. MISA/ADISA has been used by novices and experienced designers for a variety of domains ranging from well-structured to ill-structured knowledge domains (e.g., training lawyers). Paquette and colleagues (2004, 2010) found consistent improvements in both productivity and consistency of the ID products. But probably the most important result gained from these systems is the deeper understanding of ID concepts, processes, and principles. To build these systems, operational expertise in ID must be uncovered, implemented, validated, and again improved in successive versions of a system through its use in various knowledge domains.

eLearning Standards for ID

As the number of ICT-based learning platforms or authoring tools increases during the years, reusability has become more important. The goal is to enable the reuse of learning objects (or resources) in new educational contexts across a variety of e-learning delivery systems. This goal requires standard ways to describe and store learning objects or educational resources. The elaboration of international standards for learning resources has been initiated by organizations such as IMS global, IEEE-LTSC, AICC, and ISO. Duval and Robson (2001) presented a review of the earlier phases in this evolution of standards including the Dublin Core metadata initiative up to the publication of the Learning Object Metadata (LOM) standard by IEEE in 2002. Since then a host of other specifications have been published by IMS Global1. ISO has started publishing at the end of 2010 the first documents of its new Metadata for Learning Resource (ISO-MLR, 2012) standard, based on the W3C (2004) Resource Description Framework (RDF).

The work on Educational Modeling Languages (Koper, 2001), and the subsequent publication of the IMS Learning Design Specification (Griffiths, Blat, Garcia, Votgen, & Kwong, 2005; IMS-LD, 2003; Koper & Tattersall, 2004), is the most important initiative to date that integrates instructional design modeling into the international standards movement. This specification is a formal way to represent the structure of a Unit of Learning and the concept of a pedagogical method. A basic learning design involves three kinds of entities with relations between them: actor's roles, activities and environments grouping learning resources and services. Activities, performed by actors are organized in a tree structure called a method, decomposed into alternative plays, each decomposed into a series of acts, further decomposed into activity structures down to terminal learning or support activities.

IMS-LD embeds and generalizes other IMS specifications such as MD (metadata), SS (simple sequencing), CP (content packaging), RDCEO (learning objectives and prerequisites), QTI (questionnaires and tests), LIP (learner information profile) and others. SCORM, the Sharable Content Object Reusable Model supported by the ADL Technical Team (2004), can be seen as a specialization of IMS-LD to single-user simpler hierarchical activity structures. IMS-LD expands SCORM specifications in many ways:

- IMS-LD describes methods as multiactor workflow processes
- IMS-LD can provide alternative plays adapted to different target populations
- IMS-LD integrates the description of collaboration services
- IMS-LD integrates (at Level B and C) some user modeling and cross-users notifications
- Most important, IMS-LD favors instructional strategies like collaborative learning, problem solving, project-based learning, communities of practices, and multifacilitators support as found in more advanced learning strategies

With regard to the tool set, a form-based tool, RELOAD (2004), was an improvement from previously used XML editors, but it imposes too many constraints on the design process. Visual representation techniques and tools aim to free instructional designers from these constraints. Although well suited for software engineering purposes, UML graphs and diagrams, as proposed by the Best Practice and Implementation Guide (IMS-LD, 2003), pose many difficulties for instructional design. There exists more user- friendly instructional visual design software like LAMS (Dalziel, 2005), or the first MOT knowledge editors. These are useful in an inception phase, but cannot produce compliant IMS-LD executable files. This has led the construction of new visual design tools like the MOT+LD specialized editor (Paquette et al., 2005) and, more recently, the G-MOT scenario editor, the central aggregation tool in TELOS (Paquette, 2010a, 2010b).

Besides their strong influence on the standardization and interoperability of authoring tools, IMS-LD and other eLearning standards have also helped stress the importance of instructional design. IMS-LD is just a reusability format, but it has opened the spectrum of possible learning strategies that can be supported by standardized authoring tools. So the need becomes more evident for front-end methods and tools to support designers in producing high quality Learning Designs. Furthermore, the learning object paradigm has move the focus towards aggregating resources and interactions, instead of producing more text, multimedia, or Web-based document. In this new approach to ID, the learners and the facilitators are resources themselves, interacting within activities using and producing learning resources, a more cognitive and constructivist process than simple information transmission.

Social/ Semantic Web Environments

In the last decade, the now-ubiquitous Web has evolved through overlapping generations that are most of the time called the *Information Web*, the *Social Web (Web 2.0)* and the *Semantic Web (Web 3.0)*. Web 2.0 technologies are there to stay because they make the use of Internet a brand new social experience, just as the first Internet browser did 15 years ago with information access. Semantic Web technologies have the same potential to dramatically improve Web 2.0 activities that are often limited to superficial chats or simple information transmission. The new Web 2.0 and Web 3.0 technologies have an enormous potential if they are blended to support knowledge-intensive social processes.

This is now a very active research area internationally that corresponds to individuals' and organizations' needs. Here are a few research orientations that will orient the future of Web 2.0/3.0 learning environments and learning design:

1. *Modeling knowledge-Intensive social processes*. Both for work and educational scenarios, much attention is given today to multiactor workflows, but leaving aside the crucial issue of knowledge and competency acquisition that occur during these processes. On the contrary, knowledge and competency models must be at the forefront of the new learning environments to enable a transfer of competency from content experts to learners or to novice workers through collaborative knowledge exchanges. Unexplored research problems occur when the scenario or workflow is built while collaborating, in an emergent way such as in project-based learning where the learners become their own designer.
2. *Taking into account knowledge contexts of use, privacy, and trust issues in collaborative learning processes.* A huge amount of information is available for learning but it is locked from potentials users due to security and privacy concerns. These problems must be solved especially for the mobile learners whose location, device limitations, and task at hand change all the type. Context

model must be linked to task models and knowledge/competency models.

3. *Personalizing learning environments and creating more intelligent tools*. Nowadays, the abundance and popularity of Web applications, such as blogs, discussion forums, social and professional networks pose a great challenge. Web personalization and recommender systems are two important areas that attempt to cope with such information overload problems. Web personalization systems organize the Web environments based on the users' personal interests and preferences. Recommender systems suggest information, products or peer-to-peer communication in accordance with the user's personal demands and properties.
4. *Building Semantic Media User Interface*. The continued growth and importance of the Social Web has resulted in information taking many forms, including text, images, video, and more recently augmented or virtual reality environments such as Second Life. Furthermore, this information is accessible through desktop and laptop computers, and through intelligent mobile phones or tablets that bring unique constraints in terms of computing resources and user interfaces. The vast amounts of data coming out of the Social and Semantic Web entails a need for more intelligent human interfaces and visualization capabilities.
5. *Aggregating Social-Semantic tools into Learning Environments.* Data Mashups have been identified by the Horizon study (2008) as one of the leading trends for 2010–2011. Using social environments like Facebook or Wikipedia, users become Web designers, assembling text, pictures, and sound according to their needs. The issue of learning quality then comes to the forefront, while the impact of these new technologies on ID methods and tools must be investigated.

The Social and Semantic Web shapes the new learning environments, posing new challenges to Instructional Designers, fostering the need for new advances in the ID methodology and tool set. One interesting approach is to see instructional design as a knowledge-intensive collaborative multiactor process where the actors interact within a Web 2.0/3.0 environment to assemble actors, activities, and resources for learning or knowledge management.

In such a setting, personalized assistance must be given both to designers and to the user of the learning environments they produce based on semantic Web techniques, an area part of the *Adaptive Semantic Web* (Dolog, Henze, Nejdl, & Sintek, 2003) that we call *Ontology-Based Assistance Systems*. Recent research on assistance systems at LICEF (Paquette & Marino, 2011) proposes that advisor agents be grafted on environments/scenarios, built in the context of the TELOS system (Paquette & Magnan, 2008; Paquette, Rosca, Mihaila, & Masmoudi, 2006). TELOS is a service-oriented, ontology-driven system that helps build online environments for learning or for work. Its basic principle is the aggregation of resources into visual activity scenarios. In TELOS, the task model (the scenario) may represent multiactor processes or workflows integrating a variety of control patterns between tasks or activities such as splits and joins. These scenarios can be intended for any kind of actors: for engineers who aim to extend the services given by the system, for technologists who build designers' platforms, for designers who built courses or work scenarios and for the final users who interact in these scenarios.

Figure 53.4 presents the upper graph of a design process (build by an educational technologist) to help designers produce IMS-LD compliant designs: in the first activity, a designer produces the upper structure of a learning scenario (i.e., a method); in the second one, each Act in a Method is identified and defined; in the third one, a scenario model is built of each act as well as a knowledge/competency model and the association between the two structures. This third activity has a complex submodel not shown on the figure where knowledge and competencies are associated with actors, activities and resources.

When such a scenario is executed by TELOS, a Web environment is produced for the members of a design team to help them produce a learning environment model intended for learners and facilitators, to be run in the same way by the TELOS system.

Trends and ID Issues

As a conclusion, I present here four trends in methods and tools for instructional design with a set of corresponding issues that present today a challenge to the field.

From Tutoring to Open Learning Design

As shown in section "Introduction: Defining the field", at the advent of ICT in learning, it seemed natural to use ICT for the creation of learning programs. The terms CAI (Computer Aided Instruction) and CBT (Computer-Based Training) put the focus on instruction instead of learning. In this paradigm, the computer program was the teacher or a teacher aid, displaying information, asking questions and presenting more information depending on the learner's answers to previous question. Respecting the learners' pace and adapting to its answers was advocated in support for this approach. But soon, ICT in education evolved towards a more learner-oriented focus. Typically, learners would interact with computerized simulations and games, solve problems by programming the computer, search for relevant information or realize projects using software tools like text/graphic editors, database or

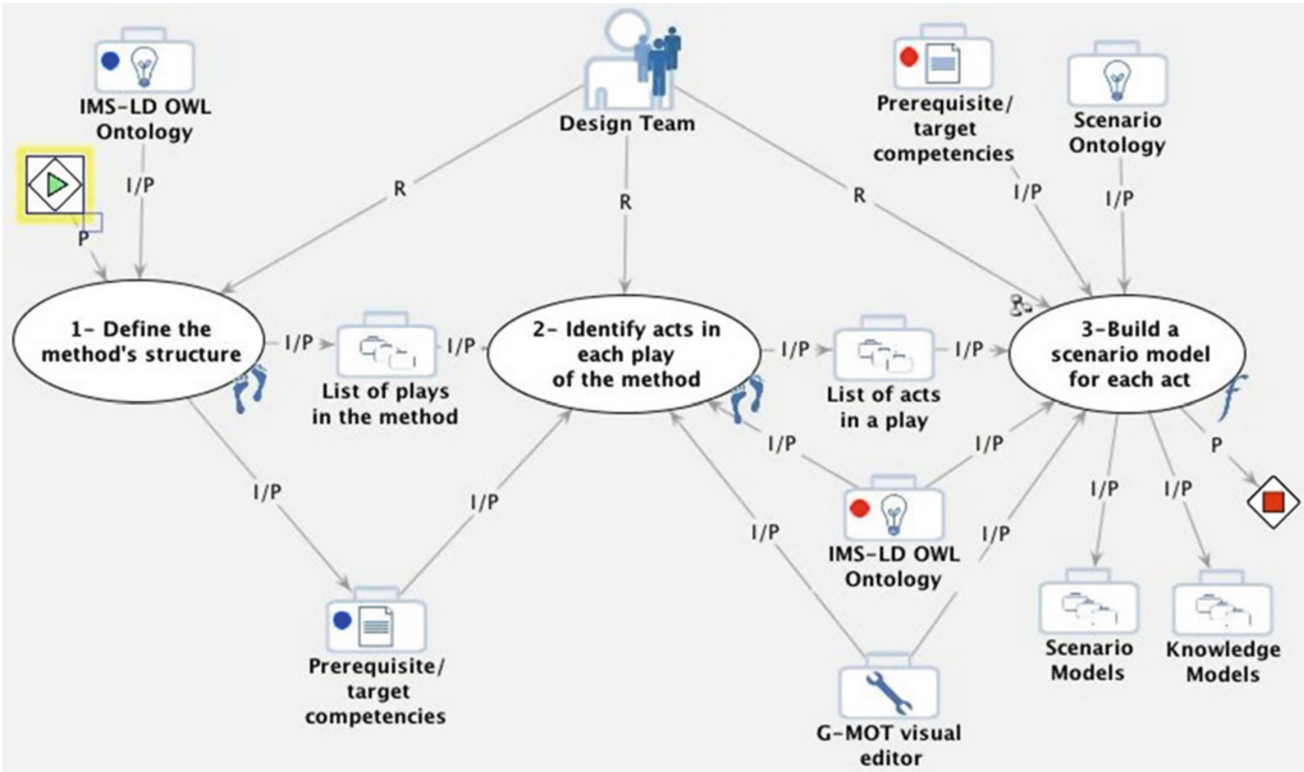

Fig. 53.4 A multiactor design scenario

spreadsheets. Nowadays, even though there are many programmed instruction courses that are useful in some cases, the trend is clearly towards more open environments where the learner uses the computer as a tool instead as a static and rigid teacher. Typically, a set of ordered activities, a scenario, is provided on the Web, where the learner is invited to find useful information on the Web, to use computer tools or to program the computer to address some question. Supporting this trend, the Web acts as a universal encyclopedia, provides a highly interactive communication system between learners and teachers, presents aggregation functions for the end user to assemble it own environment and e-portfolios.

This evolution brings to light some provocative ID issues. The first one is the challenge made to instructional design as a process distinct from delivery, some proponents even advocating the end of ID. On the contrary, others pretend that the new possibilities offered by the Web must be planned even more carefully if we want open environments to provide quality learning. Just like software engineering has brought quality that could not result from hasty coding, should not instructional engineering provide support to cope with complexity, with the larger set of decisions that face designers? But the emphasis in ID now has to shift from simply organizing information to designing activity scenarios and communication between learners and facilitators based on sound and well-proven instructional strategies and methods.

A second important issue is the quality of the information available for learning, whether the learner or teacher selects it. We are in an expanding context of billions of pages available on the Web, some providing unreliable information. On the Web, we find the good, the bad and the ugly. One solution that has been proposed is the use of learning object repositories composed of high-quality educational resources, available using metadata standardized descriptions. But this solution still has a long way to go to become mainstream.

A third issue is the support of learners in their Web-based activities. Too many times, teachers or designers will propose Web-based activities without any support, relying on the younger generation's abilities to use the Internet. Young or adult learners need support to find useful and reliable information, to learn how to communicate within the social Web, to understand the possibilities and limit of technology and their own meta-competencies in using it. Instructional designers must be supported in providing guidance on these questions, even more if the learning environment that they are planning is open and learner-centric.

From Automating to Supporting Instructional Design

Most persons designing instruction are not trained in instructional design. To address this problem, a number of researchers started building systems that could be used by inexperienced designers in their instructional design decision-making process, prior to the production stage. The general idea in the systems presented in section "Modeling Instructional Design and Job Aids" was to have a designer interact with an expert system enhanced with ID knowledge that could recommend design components to be used for the definition or production of a learning environment. So the term "automated design" seems a bit exaggerated. In fact, the design was the result an interaction between the designer and the system acting as a companion or as a tool. So the process was semiautomated. As mentioned earlier, these semiautomated systems have been used in a number of organizations where they have increased the productivity of designers and helped train new designers. Their main achievement was the production of a considerable amount of ID knowledge, but they were only marginally successful, mainly because of their complexity and their lack of flexibility and adaptivity.

These issues can be addressed by building support environments for designers in the form of mash-ups produced using workflow or scenario editors. Such editors produce executable sets of design tasks linked to tools and documents from various sources, operated by the actor(s) that perform the tasks. These scenarios can be limited in complexity, adapted to individual or team work, range from a single task to larger series of design tasks, adapted to the needs of a designer, a design team or an organization. From time to time, tasks can be reordered in the design scenario, support documents and tools can be replaced, participating actors can be added, deleted or tasks can be redistributed among actors, thus providing the needed flexibility for adaptation to a design context.

From Individual to Distributed and Collaborative ID

The first generation of instructional design tools and methods were intended for individual teachers at the design phase or in the production phase of a learning environment. Typically, an individual would sit in front of a single computer and interacts with a single software, building a design model and/or producing a CBT courseware. In more recent distance learning systems and LCMSs, the focus is also on individual designers; however, the design software is Web-based and can integrate resources available anywhere on the Web in addition to the tools provided by the LCMS. Still, the most widely used design/production environments like WebCT or Moodle do not support teamwork very well. They do not integrate an ID method. In fact, they provide generally a single set of design tasks aiming at the rapid production of a Web-based environment.

Methods like MISA and the IMS-LD specification presented above integrate a multiactor design process, taking in account the fact that in distance education and company training, the learning environments are usually designed and built by a team with members playing different roles. This links well with Web 2.0 software such as Wikipedia or GoogleDocs where documents can be built collaboratively. Flickr and YouTube offer repositories of pictures or videos to be populated by a design team. Facebook can provide some collaborative support to a design team. These social software tools must of course be integrated into design scenarios implementing parts of an Instructional Design method to produce, for example, SCORM or IMS-LD interoperable learning environments. Bringing all these elements together can provide a stimulating distributed and collaborative ID environment.

From Information-Based to Knowledge Model-Based ID

If we go back in history, preparing instruction has been mainly based on information processing. A scholar would read extensively, think a lot and synthesize large amounts of information into content documents or lectures that could be communicated to learners and novices, hopefully in a pedagogical way. Preparing lectures has been done and is still being done by most professors in much the same way, except that now the Internet provides a web of information sources. But we are now in the knowledge age where the exponential growth of available information is the rule. The use of an ever larger set of components makes the task of designing instruction much more difficult.

There are many reasons for instructional design to evolve towards ontology-based educational modeling (Paquette, 2010a, 2010b). First, within the Semantic Web framework, resources on the Internet can be described by the knowledge they support using domain ontology models. Moreover, learning environments must have a structured executable representation of the knowledge to be processed in order to help users based on their present and expected state of knowledge and competency. A third reason is that the learning process or scenario is also the result of a knowledge modeling activity using an educational modeling language. Knowledge-based ID focuses on the interaction between two models: a knowledge model of a domain (usually an ontology) that is the subject of learning and instruction, and a process model (generally a multiactor workflow or scenario) of the learning and teaching activities grouping tasks, resources used and produced by actors in the scenario. These scenario components are referenced by knowledge and competencies described in domain ontologies. Such model-based ID is necessary to cope with the inherent complexity of instructional design today, while providing flexibility and adaptability.

Conclusion

We have underlined some of the difficulties of instructional engineering, taking into account the great number of factors the designer must consider, and the constraints he must work with. Beyond the possible improvements mentioned above, it is important to develop various means of adaptive assistance for instructional engineering and to integrate them to computerized tools that support designers. This assistance cannot rest only on templates and model libraries. The implementation context must also be taken into account.

It is not easy to implement any method in an organization. It suffices to consider the time it took to convince programmers and their customers to adopt software engineering methods. The increasingly complex and vital character of information processing systems, however, provides strong arguments in favor of the adoption of such methods, making gradually anachronistic the spontaneous programming approach that marked the first decades of software production. In the field of instructional engineering, we haven't reached this point yet, although we can already see that during the next years, the same type of evolution will be increasingly necessary due to the demands of the knowledge economy. Still, ICT-based instructional engineering has a promising future for practical use in organizations. It remains also a challenging and rewarding research field.

References

Collins, A., & Stevens, A. L. (1983). A cognitive theory of inquiry teaching. In C. M. Reigeluth (Ed.), *Instructional theories in action: lessons illustrating selected theories and models* (pp. 247–278). Hillsdale, NJ: Lawrence Earlbaum.

Crowder, N. (1959). Automatic tutoring by means of intrinsic programming. In E. H. Galanter (Ed.), *Automatic teaching: The state of the Art*. New York, NY: Wiley and Sons.

Dalziel, J. R. (2005). *LAMS: Learning Activity Management System 2.0*. Retrieved from http://wiki.oamsfoundation.org/display/lams/Home

Dewey, J. (1900). Psychology and social practice. *Psychological Review, 1900*(7), 105–124.

Dick, W. (1987). A history of instructional design and its impact on educational psychology. In J. Glover & R. Roning (Eds.), *Historical foundations of educational psychology* (pp. 183–202). New York, NY: Plenum.

Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2003). *Towards the adaptive semantic web. Lecture notes in computer science*. Berlin: Springer.

Duval, E., & Robson, R. (2001). Guest editorial on metadata. *Interactive Learning Environments, 9*(3), 201–206.

Elen, J. (1998). Automating I.D.: The impact of theoretical knowledge bases and referent systems. *Instructional Science, 26*(3/4), 281–297.

*Gagné, R. M. (1985). *The conditions of learning* (4th ed.). New York, NY: Holt, Rhinehart & Winston.

Gettman, D., McNelly, T., & Muraida, D. (1999). The guided approach to instructional design advising (GAIDA): A case-based approach to developing instructional design expertise. In J. V. D. Akker, R. M. Branch, K. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 175–181). Dordrecht: Kluwer.

Griffiths, D., Blat, J., Garcia, R., Votgen, H., & Kwong, K. L. (2005). Learning design tools. In R. Koper & C. Tattersall (Eds.), *Learning design—a handbook on modeling and delivering networked education and training* (pp. 109–136). New York, NY: Springer.

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

ISO-MLR. (2012). http://en.wikipedia.org/wiki/ISO/IEC_19788

Koper R. (2001) *Modeling Units of Study from a Pedagogical Perspective, the Pedagogical Meta-Model Behind EML.* Technical Report, Educational Technology Expertise Center. Heerlen: Open University of Netherlands.

*Koper R., & Tattersall, C. (Eds.) (2004). *Learning design: A handbook of modelling and Implementing Network-based Education & Training*. Berlin: Springer-Verlag.

Landa, L. (1976). *Instructional regulation and control: Cybernetics, algorithmization, and heuristics in education*. Englewood Cliffs, NJ: Educational Technology.

Liao, M. C., Lo, S., Oyuki, M., & Wing Li, K. W. (2003). Authoring tool for ultiple Instructional strategies: Advance organizer, concept mapping and collaborative learning. *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.* Chesapeake, VA: AACE.

Locatis, C., & Park, O. (1992). Some uneasy inquiries into ID expert systems. *Educational Technology Research and Development, 40*(3), 87–94.

*Merrill, M. D. (1994) *Principles of instructional design.* Englewood Cliffs, NJ: Educational Technology Publications.

Merrill, M. D. (1998). ID Expert: A second generation instructional development system. *Instructional Science, 26*, 243–262.

Merrill, M. D., Schneider, E. W., & Fletecher, K. A. (1980). *TICCIT*. Englewood Cliffs, NJ: Education Technology Publications.

*Paquette, G. (2004) *Instructional engineering for network-based learning.* New York, NY: Pfeiffer.

Paquette, G. (2010a) Ontology-based educational modelling—Making IMS-LD visual. *Technology, Instruction, Cognition and Learning*, *7*(3–4), 263–296.

*Paquette, G. (2010b) *Visual knowledge modeling for semantic Web technology*. Hershey, PA: IGI Global.

Paquette, G., & Magnan, F. (2008). An executable model for virtual campus environments. In H. H. Adelsberger, Kinshuk, J. M. Pawlowski, & D. Sampson (Eds.), *International handbook on information technologies for education and training* (2nd ed.). New York, NY: Springer.

Paquette, G., Aubin, C., & Crevier, F. (1994). An intelligent support system for course design. *Educational Technology, 34*(9), 50–57.

Paquette, G., Rosca, I., Mihaila, S., & Masmoudi, A. (2006). TELOS, a service-oriented framework to support learning and knowledge management. In S. Pierre (Ed.), *E-Learning networked environments and architectures: A knowledge processing perspective* (pp. 79–107). Berlin: Springer.

Paquette G., Marino, O., De la Teja, I., Lundgren-Cayrol, K., Léonard, M., & Contamines, J. (2005). Implementation and deployment of the IMS Learning Design Specification. *Canadian Journal of Learning Technologies* (CJLT), http://www.cjlt.ca/

Paquette, G., & Marino, O. (2011) A Multi-actor ontology-based assistance model – A contribution to the adaptive semantic web. In S. Graf, F. Lin, Kinshuk, & R. McGreal (Eds.). *Intelligent and Adaptive Learning Systems: Technology-Enchanced Support for Learners and Teachers*, IGI Global, pp. 213–228.

*Reigeluth, C. M. (1983). *Instructional theories in action: Lessons illustrating selected theories and models.* Hillsdale, NJ: Earlbaum.

Scandura, J. M. (1973). *Strutural learning I: Theory and research*. London: Gordon & Breach Science Publishers.

Schultz, R. E. (1975). *Lesson Moniform – An authoring Aid for the Plato IV CAI System*. Retrieved from www.eric.ed.go:80/PDFS/ED127.926.pdf

SCORM, ADL Technical Team. (2004). *Online paper*. Retrieved from http://www.eife-l.org/publications/standards/elearning-standard/scormoverview/english_release

Skinner, B. F. (1954). The science of learning and the art of teaching. *Harvard Educational Review, 24*(2), 86–97.

*Spector J. M., Polson M. C., & Muraida D. J. (Eds.) (1993). *Automating instructional design, Concepts and issues.* Englewood Cliffs, NJ: Educational Technology Publications.

W3C (2004). *Ontology Web Language (OWL) overview document.* Retrieved from www.w3.org/TR/2004/REC-owl-features- 20040210/

Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. San Francisco, CA: Morgan-Kaufmann.

Winkels, R. (1992). *Explorations in intelligent tutoring and help*. Amsterdam: Ios Press.