J Braz Comput Soc (2011) 17:207–239
DOI 10.1007/s13173-011-0043-5

ORIGINAL PAPER

IMA-CID: an integrated modeling approach for developing educational modules

Ellen Francine Barbosa · José Carlos Maldonado

Received: 7 November 2010 / Accepted: 20 September 2011 / Published online: 8 October 2011

Abstract Educational modules—concise units of study, composed of theoretical and practical content, which can be delivered to learners by using technological and computational resources—are relevant mechanisms to improve the learning processes. Similar to software products, educational modules require the establishment and integration of innovative methods, tools and procedures into well-defined processes aiming at producing flexible, adaptable and high-quality products. In this sense, content modeling activity plays a fundamental role in the development of educational modules, providing a way to structure the relevant parts of the learning content. Motivated by this scenario, we propose *IMA-CID*—an approach for modeling learning content, capable of addressing conceptual, instructional and didactic issues altogether, in an integrated way. By means of a set of models, *IMA-CID* helps the author in determining the relevant parts of the learning content, providing a systematic way to structure the concepts and related information. *IMA-CID* also explores the idea of *open specifications*, providing support for the definition of dynamic contexts of learning. Besides that, the translation of *IMA-CID* models into machine-readable specifications, automatically or by hand, makes possible interoperability and promotes reusability. *IMA-CID* has been applied in the development of educational modules for different domains. The resulting modules have been evaluated in terms of the authors' and learners' perspectives. The results obtained provide preliminary evidence of the learning effectiveness, quality and flexibility achieved by the educational modules produced.

Keywords Educational modules · Content modeling · Instructional design

E.F. Barbosa (✉) · J.C. Maldonado
University of São Paulo (ICMC-USP), P.O. Box 668, São Carlos, SP 13560-970, Brazil
e-mail: francine@icmc.usp.br

J.C. Maldonado
e-mail: jcmaldon@icmc.usp.br

1 Introduction

Several initiatives in order to provide new learning opportunities and facilitate the learning process have been investigated in the last decades. One of the main challenges in this direction is how to create flexible, adaptable and high-quality educational products, capable of motivating the users (learners and instructors) and effectively contribute to knowledge construction processes in active learning environments.

The envisioned scenario is that the user can be "free" to dynamically decide which topics to navigate, progressing more or less deeply into them, according to characteristics such as course length and type, instructor's preferences, learner's profiles and learning goals. Also, there is a need for a global education, capable of crossing international, cultural and social borders in order to prepare the learners for the global market [5].

The idea of *educational modules*—concise units of study delivered to learners by using technological and computational resources—has emerged in this context. Basically, educational modules should provide: (1) transferability to different institutions and learning environments; (2) effective support to traditional learning approaches; and (3) effective support to non-traditional environments, motivating the transition from lecture-based to active and lifelong learning. Besides that, such modules should be evolvable, reusable and adaptable to different learning scenarios and objectives.

Although much effort has been put toward the definition of learning standards, models of instructional design and content modeling initiatives, there are still open issues regarding the development of educational modules to be addressed. For instance, despite the existing models of instructional design, no complete and systematic process for structuring the activities and tasks to be performed during the development of educational products has been defined. Indeed, most of such models address the fundamental activities related to the development of educational products. However, when creating an educational product, other relevant activities must be considered as complement to the fundamental ones. For instance, configuration management, documentation, coordination support, communication and infrastructure are examples of activities that should take place when developing an educational product.

Regarding the content modeling initiatives, we notice that most of the existing approaches work in a coordination level, being concerned to modeling learning designs. In short, these modeling approaches provide mechanisms to formalize the flow of activities and to identify actors, roles and resources used or produced in the learning environment, keeping open to the author the decisions regarding the content itself. Even considering the initiatives specifically designed to modeling learning content, each of them addresses different modeling issues, which can be suitable for a given learning scenario but inadequate for others. Thus, no integrated solution capable of addressing conceptual, instructional and didactic issues altogether has been explored in the modeling of learning content yet.

Motivated by this scenario, we have investigated the establishment of a systematic process for developing educational modules, aiming at providing a more complete and well-defined set of guidelines and supporting mechanisms to create, reuse and evolve them [3, 5]. $\mathcal{SP}$–$\mathcal{DEM}$ (Standard Process for Developing Educational Modules) is based on ISO/IEC 12207 standard [31], taking also in account practices from instructional design [17, 23], aspects of open development [40, 42] and of distributed and cooperative work [36].

Particularly, as part of $\mathcal{SP}$–$\mathcal{DEM}$, we have addressed three different issues of content modeling—conceptual, instructional and didactic—working in the establishment of an integrated approach for modeling learning content—$\mathcal{IMA}$–$\mathcal{CID}$ (Integrated Modeling Approach—Conceptual, Instructional and Didactic) [4].

$\mathcal{IMA}$–$\mathcal{CID}$ focuses on the development of learning content, and can be used in a complementary way with learning models working in a coordination level of abstraction. By means of a set of models, it helps the author to determine the relevant parts of the learning content, providing a systematic way to structure the concepts and related information.

The modeling approach also explores the idea of *open specifications*, providing support for the definition of dynamic contexts of learning. Considering a development/reengineering process, the $\mathcal{IMA}$–$\mathcal{CID}$ models can be especially useful to represent the instructional design rationale, playing a key role to easier evolve and maintain the resulting educational products. Besides that, the translation of $\mathcal{IMA}$–$\mathcal{CID}$ models into machine-readable specifications, automatically or by hand, makes possible interoperability and promotes reusability.

In this paper we focus on the content modeling activity and discuss the establishment of the $\mathcal{IMA}$–$\mathcal{CID}$ approach. We present the main characteristics of $\mathcal{IMA}$–$\mathcal{CID}$ and discuss the practical application of its models in the development of educational modules for different knowledge domains. The $\mathcal{IMA}$–$\mathcal{CID}$ based modules have been applied and preliminarily evaluated considering both the authors' and the learners' perspectives. In summary, we observe a very positive feedback with respect to the benefits of having a learning content well-structured and to the several different possibilities of navigation through the same content. The results obtained also provide preliminary evidence of the learning effectiveness achieved by the educational modules produced.

The remainder of the paper is organized as follows. In Sect. 2, we provide a literature review regarding standards, models of instructional design and content modeling initiatives that have been adopted in the development of educational modules. In Sect. 3, we provide an overview of educational modules, describing their main components. Also, we summarize our previous work in the establishment of $\mathcal{IMA}$–$\mathcal{CID}$, presenting a set of modeling requirements and perspectives we adopted in its proposal. In Sect. 4 we discuss the main characteristics and modeling mechanisms of $\mathcal{IMA}$–$\mathcal{CID}$ and describe the models it comprises. We also provide a set of guidelines to help instructors and/or content developers with applying the approach. Results and lessons learned from the application of $\mathcal{IMA}$–$\mathcal{CID}$ in the development of educational modules for different knowledge domains are described in Sect. 6. Finally, in Sect. 7 we summarize our contributions and discuss the perspectives for further work.

2 Background

In this section we provide an overview of the research and literature associated with the development of educational modules. Besides to provide a general view of some of the major initiatives being conducted in the area, we also intend to make the reader aware that there are open issues regarding the development of educational modules to be addressed.

In the first part of this review, we focus on the learning standardization initiatives, adopted in order to promote the development of searchable, reusable and interoperable educational modules.

In the second part, we briefly describe some of the existing models of instructional design. At the same time that these models provide an overall understanding about the fundamental activities a process for educational modules should consider, they also help on identifying the relevant activities that have not been covered by any model yet.

In the last part, we summarize the initiatives to modeling learning content. The idea is to illustrate how different are the existing content modeling initiatives and motivate the reader about the need for an integrated modeling approach, capable of addressing conceptual, instructional and didactic issues altogether, in order to create flexible, adaptable and high-quality educational modules.

2.1 Learning standardization initiatives

According to Caeiro-Rodríguez [12], the learning standardization initiatives intend to: (i) facilitate the search and location of appropriate and useful learning objects; (ii) enable the transfer of learning objects between systems; and (iii) enable the use of learning objects in different systems.

The first standardization efforts were devoted to the development of metadata schemas in order to facilitate the search of educational resources. The IEEE LOM (Learning Object Metadata) [28] is a multipart standard, developed to facilitate search, evaluation, acquisition, use and management of learning objects. The LOM data model specifies which aspects of a learning object should be described and what vocabularies may be used for these descriptions. Moreover, it defines how this data model can be amended by additions or constraints. Other parts of the standard are related to the bindings of the LOM data model, i.e., to define how LOM records should be represented in XML and RDF.

Still in the scope of learning objects metadata, we point out the Dublin Core Metadata Initiative (DCMI) [13]. DCMI intends to establish and maintain metadata standards for describing Internet resources, allowing for discovery and interoperability of resources across a range of platforms and systems. Key principles are simplicity, interoperability, extensibility and refinement. The specification is intended to be used as a core element set that may be extended for specific use.

Standards for the packaging of learning objects have been proposed to enable their transfer between systems. Since a learning object is an aggregation of several related units, an agreement is needed on resource packaging to facilitate their transfer. Packaging models define how to encapsulate all elements that made up a course or a lesson as a single unit to be easily transferred from system to system. As an example, we highlight the IMS Content Packaging specifications (IMS CP) [30], developed to promote the interoperability of learning resources between different platforms and learning managements systems. Basically, an IMS content package consists of a compressed package containing learning resources in the form of learning objects and an XML manifest that describes the content of the package.

Several standardization initiatives have also considered the organization of the content and other resources that made up the learning object to facilitate the operation of final learning systems. These specifications define how the components of a learning object should be organized in a hierarchical structure, establishing the order in which they should be delivered to learners. Such specifications are important to ensure that the same learning object may be used in different learning systems, supporting their interoperability.

SCORM (Sharable Content Object Reference Model) [1], for instance, is an implementation model for reusable learning content. It is a collection of standards, specifications and guidelines that are integrated with one another to form a single reference model. The idea is to provide a consistent framework within which learning objects can be described, assembled or package for delivery via web servers connected to learning management systems.

To sum up, standards play an essential role in the development of learning objects and, for this reason, much effort has been put toward their definition and practical application. As a consequence, learning standards have been adopted by a lot of tools and important projects. Nonetheless, there is still a lot of space for improvement and even redesign. For instance, although the existing metadata developments have established a solid framework for describing technical and physical attributes of learning objects, there are still issues that remain opened for investigation. It is the case of metadata and dynamic annotation. Memmel et al. [43] pose the idea that learning objects must be watched as they are used. Since current metadata concepts do not support dynamic annotation, the authors claim that the future of learning has to bring, among other things, extensions of metadata concepts that go far beyond the current conventional approaches.

As a final remark, even though our work does not focus on the research issues directly related to learning standards, it is important to highlight that the learning content we have produced is aligned with such standards, particularly with those widely adopted and supported by the learning community.

2.2 Models of instructional design

The most basic model of instructional design is the ADDIE model [17, 23]. ADDIE is the acronym for *Analysis*, *Design*, *Development*, *Implementation* and *Evaluation*, which correspond to the five stages of the model. The model begins with an analysis of instructional needs and solutions, followed by the design and development of learning objectives and methodologies, implementation of the learning content, and a summative evaluation of the resulting product.

One of the strengths of the ADDIE model is that it offers a series of questions to ensure a critical examination of instructional goals, learning objectives and learner needs at each stage of the design process. The model proceeds from one stage to another with revision occurring throughout the design process to ensure that the product of design does not run askew from the instructional goals [22]. ADDIE is considered the starting point to derive specific models for developing educational products. Although several models for instructional design have been developed, most of them are still based on the core ideas of ADDIE model.

Besides ADDIE, there are several other models of instructional design. The CLE (Constructivist Learning Environment) [32], for instance, focuses on authentic learner problem-solving model. The Jonassen's model conceives of a meaningful problem, question or project as the focus of the environment, surrounded by interpretative and intellectual support systems such as related cases and information resources; cognitive, conversation and collaboration tools; and social context that support learner problem solving [32]. The learner's goal is to interpret and solve the problem or complete the project. Basically, CLE establishes a list of learning activities that students should perform (*Exploration*, *Articulation* and *Reflection*) and a list of instructional activities that the environment should provide in order to support the learners (*Modeling*, *Coaching* and *Scaffolding*).

HDM (Hypermedia Design Model) [42] is a constructivist model of design created for the Web and other hypermedia environments. The model comprises six stages. The first two stages (*Define your learning domain* and *Identify cases within the domain*) define the instructional content, goals and format. HDM then splits into two paths [16]: (1) in the guided path, HDM provides suggestions to the learner as to the design goal (*Identify themes and perspectives*) and includes multiple paths to follow (*Map multiple paths link cases*); (2) in the learner-controlled path, learners are able to specify their own learning objectives and are able to navigate a path of their own creation (*Provide learner-controlled navigation through cases*). The final step in HDM (*Focus learner self-reflection*) is to encourage learner self-reflection in order for learners to determine if their learning objectives have been achieved.

Aiming at applying the principles of objectivist and constructivist design, Farrell & Carr [22] proposed the Learning Object Design Model. In short, the model implements the strengths of the ADDIE, CLE and HDM models by integrating ADDIE's comprehensive and systematic approach to design, CLE's focus on relevant and engaging problem solving, and HDM's provision for learner control and design guidance.

LODAS (Learning Object Design and Sequencing Theory), proposed by Wiley [68], addresses the issues of granularity (scope and design) and sequencing (combination) for developing learning objects. LODAS was designed to support the instructional use of learning objects and facilitate a significant amount of reusability across objects. By combining a number of existing instructional design theories, including Elaboration Theory [53]), Work Model Synthesis [24], Domain Theory [68], and the Four-Component Instructional Design model [45], LODAS provides taxonomy and design guidance for different types of learning object. Wiley's proposal is composed of six steps [68]:

1. *Preliminary activities*: In this step the designer needs to determine the appropriateness of using the LODAS for achieving the organization or course goals.
2. *Content analysis and synthesis*: In this step the designer needs to: (a) identify the necessary cognitive skills to achieve the overall goal of instruction; (b) break larger tasks into their associated smaller components, getting simpler as the decomposition continues until no more decomposition is possible; and (c) synthesize work models, i.e., the constituent skills are recombined into activities that people perform in the real world.
3. *Design practice and information presentation*: In this step the designer needs to identify the practice and instruction necessary for each task.
4. *Learning object selection or design*: In this step the designer needs to: (a) review preexisting learning objects available in metadata repositories; and (b) create new learning objects.
5. *Learning object sequencing*: In this step the designer needs to sequence educational resources based on their cognitive complexity.
6. *Loop back for quality improvement*: In this step the designer has finished the instructional design and the development of learning situation and then he/she starts a process of quality improvement, which should become an ongoing activity. Formative and summative evaluations can be developed to ensure quality improvement.

More recently, grounded in the idea of OER (Open Educational Resources) [54, 62, 64], i.e., educational materials purposely made available for free use for others, OER development approaches have also emerged. McGreal [41], based on Extreme Programming (XP) methodology, discusses 13 practices adapted from software engineering to aid in the course development process. Leinonen et al. [34, 35] describe a process of research-based design aiming at developing new learning technologies. The process is divided into four iterative phases (*contextual inquiry*, *participatory design*, *product design*, and *production of software as hypothesis*), which happen partly in parallel. According to the authors, the process resembles a hermeneutic circle where all research and design operations increase the researchers' and the designers' understanding of each other and the context.

By analyzing the models of instructional design described, we notice that most of them address the fundamental (primary) activities related to the development of educational products. However, when applying a specific instructional model in the practice, many other relevant activities must be considered as complement to the fundamental ones. Supporting activities (e.g., Configuration Management, Documentation) and organizational activities (e.g., Coordination, Communication, Infrastructure) are examples of activities that should take place when developing an educational product.

Despite the several models of instructional design, no complete and systematic process for structuring the activities and tasks to be performed during the development of educational products has been defined so far. Indeed, as important as the selection of the appropriate model of instructional design is the definition of the adequate supporting technologies and tools to be used, the human resources and their roles in the development, as well as the deliverables (i.e., the precisely described products of this process).

In this sense, we argue that there is a lack of well-defined processes for instructional design, capable of defining a complete and systematic way to produce flexible, adaptable and high-quality educational products. This scenario motivated us to work on the establishment of $\mathcal{SP}$–$\mathcal{DEM}$, briefly described in Sect. 3.2.

2.3 Content modeling initiatives

All models of instructional design point out the need for structuring and organizing the learning content. In general, the establishment of models for representing learning content involves several different aspects. For instance, we have to consider the specific characteristics related to the knowledge domain, to define the practical tasks and the evaluation mechanisms that will be applied to learners, and to establish pedagogical sequences for presenting the information. Besides that, the content modeling activity can take place in different levels of abstraction, from coordination to instructional and pedagogical levels.

EMLs (Educational Modeling Languages) [21] have been proposed to support the description of instruction mechanisms and resources used in the learning. According to Villani [66], they provide a meta-model that enables to capture the resources (e.g., texts, figures and tools) used during the instruction as well as the instructional design information that establish in which manner such resources are intended to be used.

An EML focus on the coordination of the entities (e.g., persons, documents, tools) involved in instruction instead on the pedagogical approaches or instruction elements. Indeed, the main goal of EMLs is to support the modeling of the coordination issues between such entities (e.g., the documents/tools that can be accessed/used by a learner) together with the establishment of particular goals responsible for driving and controlling the way in which such entities are intended to participate and interact. To allow this kind of modeling, EML meta-model uses to be arranged according to an activity scheme involving three main entities: (1) the *Goals* that have to be achieved in each *Activity*, which are usually related with an *Object* to be produced; (2) the *Subject(s)* that have to carry out each *Activity*, who participate playing specific *Roles*; and (3) the *Environment* where each *Activity* has to be carried out.

Following the basic EML activity scheme, Learning Design specification (LD) [29] provides a notation to support the description of instruction in computational environments. Such notation is expressed using XML tags that must be arranged in accordance with the LD meta-model. As pointed out by Paquette et al. [50], the LD specification leaves open the choice of instructional models and tools that can support designers in the development of educational products.

The main problem related to LD specifications is the lack of explicit support of the instructional design rationale. Thus, LD uses low level coordination mechanisms to describe the coordination of the learning elements, but it does not explicitly capture the coordination rationale involved in the instructional design. Villani [66] argues that this problem can be solved considering a similar solution as in computer programming languages, where high-level languages and assembler languages are focused on different concerns. EMLs have already been considered as assembler languages (e.g., LD) and high-level educational modeling languages remains to be developed yet.

In a related perspective, Rodríguez-Artacho [55] introduced the PALO language as a cognitive-based approach to EMLs. Basically, the PALO language provides a layer of abstraction for the description of learning material, including the description of learning activities, structure and scheduling. The language is adherent with a reference framework to describe learning materials [56]. Such framework makes use of domain and pedagogical ontologies as a reusable and maintainable way to represent and store learning content, and to provide a pedagogical level of abstraction in the authoring process.

MISA (French acronym for *Méthode d'Ingénierie des Systèmes d'Apprentissage*) [50] is an instructional engineering method supporting 35 main tasks or processes and some 150 secondary tasks. The method is based on a problem-solving approach, comprising six phases: (1) identify the educational problem; (2) define preliminary solution; (3) build learning system architecture; (4) design instructional materials; (5) model, produce and validate materials; and (6) prepare delivery of learning system.

In each of phases 2 to 6, MISA proposes the development of four axes. The *Knowledge Model* enables the identification of various types of link and knowledge, including the competencies to be developed. Shortly, the model embodies the following types of knowledge: (1) *conceptual knowledge*—allows the description of the objects of the designer must use or produce; called documentation elements, such objects make up the blueprint and specifications for a learning system; (2) *procedural knowledge*—aims to define the actions to be performed on these objects, or how the designer achieves each task necessary to the production of the training system; (3) *strategic knowledge*—consists of statements to help the designer decide when or why to make a particular choice during the application of a certain procedure. Additionally, the concept of *competence* is reconciled with concepts of knowledge, skills, and learning needs; skills are classified enabling integrated processing of cognitive, affective, social and psychomotor aspects.

The *Instructional Model* guides the creation of learning units. It is responsible for the identification of a set of instructional strategies describing the learning activities, offering a precise and broad definition of the central concept of the instructional scenario. Paquete et al. [50] observed that MISA's specification of an *Instructional Model* can be seen as a kind of EML.

The *Learning Model* makes it possible to carry out the macro-design of the instructional materials without prejudging the decisions that will be taken later by specialists as the various media are built (micro-design).

The *Delivery Model* covers the delivery infrastructures and the training management tasks and processes necessary to access the learning system.

MISA involves the interaction of many specialists such as content experts, instructional designers, media producers and training managers. Each of these main actors is central to one of the four axes, but they all interact and intervene in all axes as well.

According to Paquete et al. [50], MISA deals with modeling of learning designs in the sense it formalizes the flow of activities and precisely identifies the actors, their roles as well as the resources used or produced in the environment. Furthermore, the authors claim that LD specification and the MISA method complement each other. The LD specification provides a standardized formal and machine-readable representation of learning design, whereas MISA proposes a systemic and systematic method to design and implement such learning designs.

Finally, there are some approaches specifically designed for developing educational hypermedia applications. DAPHNE (Portuguese acronym for *Definição de Aplicações Hipermídia na Educação*) [33] is based on the Concept Mapping Theory [48] and on the Information Mapping Technique [27]. EHDM (Educational Hyperdocuments Design Method) [49] is based on the Concept Mapping Theory and on the Michener's work [46]. MAPHE (Portuguese acronym for *Metodologia de Apoio a Projetos de Hipertextos Educacionais*) [51] incorporates to the Concept Mapping Theory some usual relationships of the Object-Oriented Modeling [57].

By analyzing the content modeling initiatives described, we notice that EMLs, PALO language and MISA focus on coordination issues. In short, these modeling approaches are concerned to modeling learning designs, keeping open to the author the decisions regarding the content itself. Since the focus of $\mathcal{IMA}$–$\mathcal{CID}$ is to model the content details (fine-grained modeling approach), we can explore its adoption in a complementary way, especially considering the MISA approach.

On the other hand, DAPHNE, EHDM and MAPHE focus on the modeling of learning content; hence, in the same direction of $\mathcal{IMA}$–$\mathcal{CID}$. However, each approach addresses specific issues of content modeling. Conceptual aspects, for instance, are emphasized in DAPHNE and MAPHE while EHDM just provides mechanisms to support the domain modularization. Instructional issues are addressed by DAPHNE and EHDM; MAPHE does not provide specific mechanisms for dealing with them. Didactic aspects are considered in all approaches by means of precedence relations, but no support for dynamic contexts of learning are provided.

Motivated by this scenario, we are interested in an integrated approach for modeling the learning content, capable of providing a complete set of models to address the conceptual, instructional and didactic perspectives altogether. Also, we intend to investigate a way to represent dynamic contexts of learning, where the elements of the content can be dynamically determined according to specific parameters defined in terms of the characteristics of the course, learners and instructors. Such characteristic is important to foster aspects of customization and adaptability of the educational products in order to better engage the learners (and teachers as well) in an active learning process. Besides that, we intend to explore the translation of the graphical representations of the learning content into machine-readable specifications (as already occurs at the coordination level), as a way to promote interoperability and reusability. These research points have not been considered by the existing modeling approaches at the content learning level, motivating us to the proposition of $\mathcal{IMA}$–$\mathcal{CID}$, described in Sect. 4.

3 Developing educational modules

In this section we describe the main components of an educational module and its main characteristics. An overview of the mechanisms we have investigated to support its development is also presented.

3.1 Educational modules: a definition

Educational modules are concise units of study, composed of theoretical and practical content, which can be delivered to learners by using technological and computational resources [3]. Figure 1 illustrates the main components of an educational module.

For theoretical content, instructors use books, papers, web information, slides, class annotations, audio, video, and so on. Practical content is the instructional activities and associated evaluations, as well as their resulting artifacts (e.g., executable programs, experimental studies, collaborative discussions). Specific tools related to the subject knowledge domain and the results obtained from their application can also be seen as practical content. Such tools can be integrated as part of an educational module in order to enable the application of fundamental concepts in realistic scenarios. This integration fosters training situations and promotes exchange of technology between industry and academia, also providing learners with domain-specific skills.

Theoretical and practical content are integrated in terms of learning materials. In order to deliver the materials to learners, an adequate infrastructure is also needed. Learning environments (e.g., *WebCT* [25], *Blackboard* [8], *Moodle* [20], *Sakai* [59], *DotLRN* [19]) as well as technological and computational resources, such as mechanisms to transform the content of traditional lectures into searchable and extensible digital media (e.g., *e-Class* [11, 52]) and to support collaborative work and augment communication and discussion among instructors and learners (e.g., *CoWeb* [18]), illustrate some of the required infrastructure related to the educational modules.

It is also important to define an educational module in terms of a learning object. In a very broad definition, IEEE/LTSC states that a learning object corresponds to "*any entity, digital or non-digital, that can be used, reused or referenced during technology supported learning... Examples of learning objects include multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning*" [28].

Considering the IEEE/LTSC's definition of learning objects, both theoretical and practical content can be seen as learning objects. Since educational modules are composed of theoretical and practical content, they can also be defined as a collection of learning objects. Moreover, learning objects can be hierarchically aggregated and represented in different granularities, i.e., a learning object can also be seen as a collection of learning objects. Therefore, an educational module can be defined as a learning object. In this paper we use the term *educational module* with the same meaning of *learning object*.

The development of educational modules should take in account some key characteristics of knowledge. We have to consider, for instance, the knowledge structure and organization, i.e., how the information related to the subject domain can be integrated and how to establish a well-defined structure to represent it. The way the educational module is structured and organized deeply impacts its learning effectiveness. Hence, it is fundamental that when creating an educational module the authors (not necessarily the domain experts) clearly understand the subject domain, being able to identify and organize concepts and relevant information

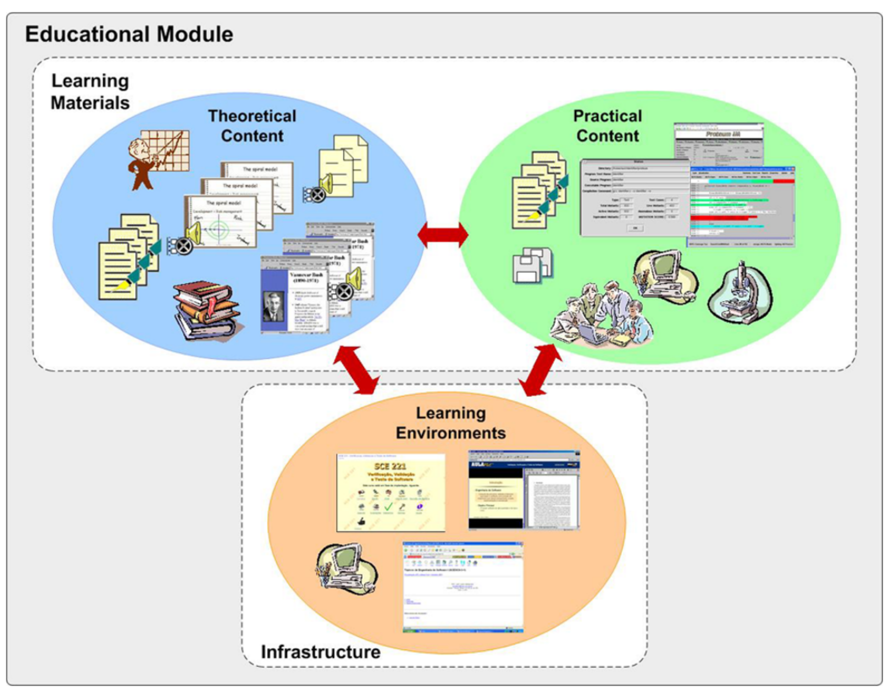


Fig. 1 Main components of an educational module

and, also, to specify practical activities and related evaluations.

Other characteristic to be considered refers the dynamic and evolutionary aspect of knowledge, i.e., new knowledge is continuously produced and referenced in consequence from previous learning experiences. In each class, traditional, blended or distance one, new content (e.g., slides, annotations, texts, results and sub-products from practical activities) is created when delivering the module to learners, and should be incorporated into the content previously defined. In fact, the learning content is continuously expanded (active growth) in consequence from the contributions of all participants (learners and instructors) of the course. Furthermore, such content is frequently referenced (intrinsic reference), aiming at both consolidating the previous knowledge acquired as well as motivating the apprenticeship of new concepts and information inter-related.

Finally, issues regarding knowledge reuse and sharing should also be addressed. Reusability allows the content developed in a given learning context to be easier available and transferable to another one, with different educational purposes. Nevertheless, different needs, backgrounds and skills can represent a barrier to effective learning. Since there can be widely different viewpoints and assumptions regarding the same subject matter, the consequent lack of a shared understanding can lead to poor communication and collaboration, impacting the learning processes in general. Indeed, a shared conceptualization of the knowledge domain represents the basis for developing and for reusing high-quality educational modules.

3.2 A process for developing educational modules

Similar to software products, educational modules require the establishment and integration of innovative methods, tools and procedures into systematic processes aiming at producing flexible, adaptable and high-quality products. The development of such modules can involve developers from different domains, working on multi-disciplinary and heterogeneous teams, geographically dispersed or not. They should cooperate, sharing data and information regarding the project. Furthermore, we should consider the adoption of supporting tools, which can be used either as part of the educational module under construction or as a mechanism to automate its development process.

Considering this scenario, we have investigated the standardization of processes for developing educational modules [3, 5]. The Standard Process for Educational Modules ($\mathcal{SP}$–$\mathcal{DEM}$) is based on the International Standard ISO/IEC 12207 [31], tailored to the context of educational modules by including aspects of content modeling [3, 4], practices from instructional design [17], and issues of distributed and cooperative work [36].

In short, $\mathcal{SP}$–$\mathcal{DEM}$ establishes a set of processes that can be employed to acquire, supply, develop, deliver, operate, and maintain educational modules. Three categories of processes are defined: (1) the primary processes deal with the main activities and tasks performed during the life cycle of an educational module; (2) the supporting processes support other processes and contribute to the success and quality of the development project, and (3) the organizational processes are employed by an organization to establish, implement and improve an underlying structure made up of associated life cycle processes and personnel. Figure 2 shows the general structure of the standard. Dashed rectangles are the processes adapted from the ISO/IEC 12207 [31]. Dotted rectangles are the processes adapted from the standard process for geographically dispersed working groups [36]. White rectangles are the processes specifically developed to the context of teaching and learning.

$\mathcal{SP}$–$\mathcal{DEM}$ is responsible for the establishment of a unique development structure to be adopted and followed by the entire organization [3, 5]. However, changes in organizational procedures, educational paradigms and principles, learning requirements, development methods and strategies, as well as the size and complexity of the projects, among other aspects, impact the way an educational module is produced. In this sense, to be used in particular projects, the processes should be defined case by case, taking into account the specific features of each project.

Process specialization and instantiation have also been explored in order to apply the standard process into specific learning environments and organizations. Basically, the definition of a process for developing a given educational module should consider its adequacy to: (1) the involved technologies, supporting mechanisms and budget; (2) the domain of the educational application; (3) the characteristics of the module; (4) the maturity level of the development team; and (5) the characteristics of the organization. As a consequence, processes into different levels of abstraction are defined. More detailed information about $\mathcal{SP}$–$\mathcal{DEM}$ definition, specialization and instantiation can be found in [3, 5].

3.3 Content modeling requirements

As mentioned before, content modeling plays a key role in the development process of educational modules [12, 39, 56]. Basically, it helps the author in determining the main concepts to be taught, providing a systematic way to structure the relevant parts of the subject knowledge domain. Besides that, how the content is structured and organized directly impacts the reusability, evolvability and adaptability of the module.

The establishment of models for representing learning content involves several different issues. For instance, we have to consider the specific characteristics related to the

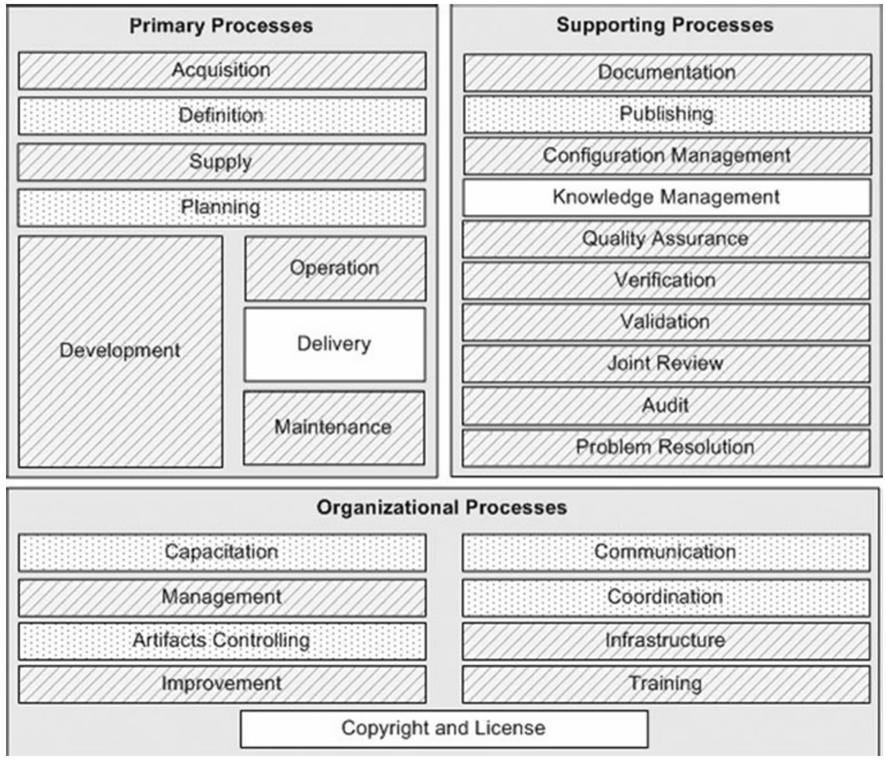


Fig. 2 Main structure of $\mathcal{SP}$–$\mathcal{DEM}$

knowledge domain, to define the practical tasks and the evaluation mechanisms that will be applied to learners, and to establish pedagogical sequences for presenting the modeled information.

As discussed in Sect. 2, the existing modeling approaches work in different levels of abstraction; some of them address learning coordination issues while others specifically deal with the content development itself. Besides that, even considering only the modeling initiatives in the same level of abstraction, we noticed that each approach addresses specific issues; thus, since there is not a set of predefined requirements for content modeling, each model deals with different perspectives which can be suitable for a given learning scenario but inadequate for others.

Motivated by this scenario and based on the typical features required by educational modules, we proposed a preliminary set of content modeling requirements [3]. Nine requirements were defined as part of our set: (1) concept taxonomy; (2) concept composition; (3) domain-specific relationships; (4) hierarchical decomposition; (5) knowledge categories; (6) pedagogical order; (7) learning contexts; (8) history mechanisms; and (9) event propagation (broadcast mechanisms). They are briefly described next.

1. *Concept taxonomy*: According to Ausubel's learning theory [2], the meaningful learning takes place by incorporating new concepts and propositions into the learner's previous knowledge framework. In this sense, the establishment of hierarchical structures of the knowledge domain can enhance the learning process and promote the integration among learner's previous and new knowledge.

   Concept taxonomy relationships, in which the more inclusive and general concepts are identified and represented prior to the more specific ones, can be useful for establishing hierarchical structures, helping to define and organize the relevant concepts associated with the knowledge domain.
2. *Concept composition*: Some concepts are better understood when "broken" into small pieces. The idea of concept composition indicates that a concept is compounded by other ones and can be explored to facilitate the comprehension of the knowledge domain.
3. *Domain-specific relationships*: Both concept taxonomy and concept composition represent generic categories of relationships, applicable to any kind of knowledge domain. To address the details and particularities of each different knowledge, domain-specific relationships should also be represented. Domain-specific relationships have their meaning associated to a particular subject, carrying their own semantics. In other words, they represent specific relations, user-defined, whose interpretation depends on the domain (or application) subject to modeling.
4. *Hierarchical decomposition*: The hierarchical structure of the knowledge domain discussed in Ausubel's learning theory [2] also implies the idea of hierarchical decomposition (or modularization) of the knowledge. Ac-

tually, if the number of relevant concepts associated with the knowledge domain is high, their representation in a unique space may compromise the legibility of the content model and affect the quality of the material under development. So, mechanisms for structuring large bodies of knowledge into several smaller groups of concepts are fundamental. Notice that each group contains only concepts strictly related, characterizing a cohesive unit of study. Also, general (basic) units should be presented prior to specific ones.

The idea of hierarchical decomposition allows us to represent the knowledge domain under different abstraction levels. The lower is the level more are the details which can be explored in terms of structure of concepts.

5. *Knowledge categories*: Different kinds of information can be categorized as part of the knowledge domain or as a mechanism to facilitate its understanding. Several theories and techniques are referred to support the establishment of knowledge categories [27, 44, 46]. Merrill's Component Display Theory [44], for instance, divides the information into *concepts*, *facts*, *procedures* and *principles*. Michener's technique [46] specifies *concepts*, *results* and *examples* as the basic elements for structuring mathematical knowledge. Horn [27] proposed the Information Mapping Technique, which consists of dividing the information related to the nodes of a conceptual map into small portions of information, called *information maps* (*concept*, *structure*, *procedure*, *fact*, *process*, *classification*, and *principle*). Similarly, information maps can be divided into smaller parts, called *information blocks* (*definition*, *example*, *classification list*, *rule*, *synonym*, *theorem*, *exercise*, and so on).

   We highlight that there is a vast literature on Artificial Intelligence exploring the topic of knowledge representation as well. However, providing a deep view on the related theories and techniques is out of the scope of this paper. Readers interested in the foundations of knowledge representation can see [10]. For a more current perspective, also addressing issues of knowledge in learning, see [58].

   Actually, regardless of the theory or technique adopted, the main issue is to provide adequate mechanisms to specify and differentiate the information, avoiding inconsistencies and/or ambiguities. At the end, the establishment of knowledge categories is an important support to qualify the elements of the learning content in terms of their roles and instructional purposes.

6. *Pedagogical order*: The most basic way for representing the didactic dependencies related to a knowledge domain is to define pedagogical orders among its components. The idea is simple: if an information is prerequisite for another one, the former must be studied before the latter, establishing a sequence of presentation for them. A pedagogical order may characterize a simple "preference" for presenting the information or can be "mandatory", pedagogically necessary for the effectiveness of learning. Also, notice that more than one pedagogical order can be established for the same information, varying according to the learner's profile, instructor's preferences, learning goals, course length, among other aspects.

7. *Learning contexts*: Learning contexts are defined to allow learners with different profiles and goals to explore the same material through distinct perspectives. Learning contexts and pedagogical orders are intrinsically related—each sequence of presentation established among the components of the knowledge can be seen as a specific learning context. Notice that alternative ways to access the information (e.g., guided tours, indexes, learning scenarios) constitute supporting mechanisms to the establishment of learning contexts and should be considered when modeling the learning content.

8. *History mechanisms*: Sometimes an active learning context needs to be suspended and restored later. For instance, suppose a learner who is navigating through a theoretical content and, at a given point, he/she is required to practice the theory by doing an exercise. Clearly, there is a context change since the learner has now to navigate through problems and exercises instead of concepts and theoretical information. After finishing the proposed exercise, the learner must go back to the original context, and restart to explore the theory related to the knowledge domain. History mechanisms, as those proposed by Harel in the statechart definition [26], are suitable for modeling such situation.

   History mechanisms are also useful to "remember" the paths the learner has traversed when exploring the material. As a consequence, assessment and evaluative aspects can be explored as well.

   Finally, history mechanisms can reduce the number of transitions required to modeling the learning content, reducing the complexity for representing the knowledge domain. In fact, since the number of concepts and interrelated information is high, the absence of history mechanisms can lead to an explosion of the number of transitions (relationships) represented in the pertinent models.

9. *Event propagation* (*broadcast mechanisms*): Learning content comprises different kinds of media (e.g., audio, video, animation, text, graphic, image). A typical problem related to this characteristic is the lost of synchronism among these elements, which can affect not only the quality of the learning material but also the performance of the learner when using it. Aspects of event propagation and broadcast mechanisms, also explored in the statechart definition [26], are relevant to be considered in order to guarantee the concurrency and synchronism for the module under development.

We have also worked on the identification of modeling perspectives in order to characterize adequate models for better representing the elements of the learning content. In short, learning content can be defined based on the description of concepts and other relevant information of the knowledge domain, together with additional elements (examples, further explanations, exercises, problems to be solved, suggestions for further study, evaluations, and so on) [39]. In this sense, three different perspectives were identified [3, 4]: (1) the *conceptual* perspective refers to the main concepts description; (2) the *instructional* perspective deals with additional information used to perform the learning process; and (3) the *didactic* perspective aims at relating the conceptual and instructional objects, providing facilities to establish a sequence for presenting them.

Based on the modeling requirements and perspectives, we carried out a series of comparisons involving the existing modeling approaches [3]. The goal was to identify strengths and weaknesses of each of them. In general, while some approaches seem to be particularly interesting in addressing conceptual issues, others deal with relevant elements under the instructional perspective, and other ones demonstrate an expressive power on representing didactic aspects. However, none of them provides a complete and integrated set of features addressing the conceptual, instructional and didactic perspectives altogether. These observations provide evidence of the need for integrated modeling approaches, capable of putting together in a unique proposal the variety of aspects and issues that should be addressed.

Table 1 Perspectives and requirements for content modeling

Modeling perspectives	Modeling requirements
Conceptual	Concept taxonomy
	Concept composition
	Domain-specific relationships
Instructional	Hierarchical decomposition
	Knowledge categories
Didactic	Pedagogical order
	Learning contexts
	History mechanisms
	Event propagation

Finally, we established a connection between the modeling perspectives and the modeling requirements. Table 1 illustrates this matching. The first three requirements are relevant under the conceptual perspective, the fourth and fifth ones are related to the instructional perspective, and the others refer to the didactic perspective. Actually, this connection was the starting point to the definition of the models for representing the learning content and their integration into the $\mathcal{IMA}$–$\mathcal{CID}$ approach. $\mathcal{IMA}$–$\mathcal{CID}$ is described next. More details about the modeling requirements and perspectives we proposed are available in [3].

4 $\mathcal{IMA}$–$\mathcal{CID}$: an integrated approach for modeling educational content

Based on the modeling requirements and perspectives discussed in the previous section, we proposed $\mathcal{IMA}$–$\mathcal{CID}$ (*Integrated Modeling Approach—Conceptual, Instructional and Didactic*) [3, 4]—an integrated approach for modeling learning content, composed of a set of models, each one addressing specific issues (conceptual, instructional and didactic) to support the development of learning content. Figure 3 summarizes the key points of our proposal.

4.1 Conceptual model

The *Conceptual Model* consists of a high-level description of the knowledge domain, representing its main concepts and the relationships among them. The relationships can be divided into two classes: structural and domain-specific. Structural relationships are useful to set up taxonomies among concepts and make inferences about the

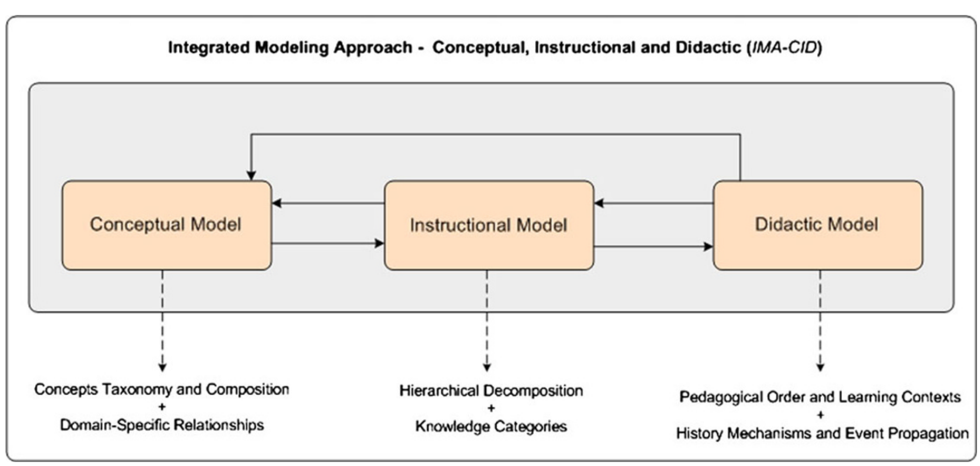


Fig. 3 The $\mathcal{IMA}$–$\mathcal{CID}$ modeling approach

knowledge, representing a generic category of relationships, applicable to any kind of domain. Relations such as *type-of* and *part-of* are examples of structural relationships. On the other hand, domain-specific relationships are user-defined and have their meaning associated to a particular subject, carrying their own semantics. That is, they represent specific relations, whose interpretation depends on the domain being modeled.

To construct the conceptual model, we focused on the Conceptual Mapping Technique, proposed by Novak [48]. Among the reasons for choosing this technique we point out: (1) it is suitable for representing concepts and for structuring the knowledge domain; (2) it is intuitive and easy to use; (3) it is based on educational principles, having a good acceptance among educational specialists and professionals; and (4) it is adopted by the majority of existing modeling approaches for learning content. In addition to the rules for creating conceptual maps, we included some specific notations aimed at representing the relationships of concept taxonomy (*type-of*) and concept composition (*part-of*).

4.2 Instructional model

Besides concepts, information items and instructional elements should also be considered as part of the knowledge domain. In the *Instructional Model* we are interested in defining such additional information related to the concepts previously identified. Notice we are not interested in how the information will be associated, but in what kind of information we can use to develop more significant and motivating learning content.

The construction of the instructional model involves two phases: (1) the refinement of the conceptual model; and (2) the definition of the instructional elements. In the first phase we specify what kind of additional information can be incorporated to the concepts already represented in the conceptual model. We call them *information items*.

As we mentioned before, several theories and techniques can be referred to support the establishment of information items [27, 44, 46]. In our work we adopted the Component Display Theory (CDT), proposed by Merrill [44]. Regarding content, CDT specifies the following elements:

- *Concepts*: Symbols, events and objects that share characteristics and are identified by the same name. Concepts make up a large portion of language and understanding them is essential to communication.
- *Facts*: Logically associated pieces of information. Names, data and events are examples of facts.
- *Procedures*: A set of ordered steps to solve a problem or accomplish a goal.
- *Principles*: Elements that work through either cause-and-effect or relationships. They explain or predict why something happens in a particular way.

Among the reasons for choosing the Merrill's theory we point out: (1) it has been widely used for structuring and representing "pieces of information"; (2) it is simple and easy to use (Horn's Information Mapping Technique, for instance, divides the information into several different *information maps*, which are divided again into parts even smaller, called *information blocks*); and (3) it can be used for structuring any type of knowledge domain (Michener's technique, for instance, is more suitable for structuring elements of the mathematical knowledge).

It is important to highlight that we did not prescribe the use of Merrill's knowledge categories as mandatory. For instance, the author could adopt Michener's categories [46], structuring the information content into *concepts*, *results* and *examples*. Actually, the author is free for choosing the knowledge categories he wants. The flexibility of choosing the knowledge categories to be represented in the instructional model aims to guarantee the model to be independent of particular learning theories and/or principles, which can be defined by the author.

In the second phase we define the *instructional elements*, used to complement the information items. Three types of element are identified [3, 4]:

- *Explanatory elements*: Deal with the complementary information used for explaining a given topic—examples, hints, suggestions of study, and so on. They can play different roles depending on their purpose. An example, for instance, can be associated according two distinct perspectives—to motivate the study of the topic, or to illustrate its use.
- *Exploratory elements*: Allow the learner to navigate through the domain, practising concepts and other relevant information. Guided exercises, simulations and hands-on assignments are representative of this category.
- *Evaluative elements*: Allow to assess the learner's proficiency on the domain. Diagnostic, formative and summative evaluations, in terms of subjective and/or objective questions, are examples of evaluative elements.

At this point it is also important to define the media (continuous and discrete elements) to be related to information items and instructional elements. Indeed, the establishment of adequate media, specially the continuous ones, is fundamental for developing richer interactive content, capable of motivating the learners and effectively contributing to their knowledge construction processes in learning environments.

As a support to construct the instructional model, we adopted the HMBS (Hypertext Model Based on Statecharts) model, proposed by Turine et al. [63]. The HMBS definition is the following.

A hypertext H is a 6-tuple $H = \langle ST, P, R, M, L, V \rangle$, in which [63]: ST is a statechart structure; P is a *set of pages* corresponding to the pieces of information included in the application; R is a *set of readers*, or channels of presentation, which are abstractions used to specify the requirements involved in the presentation of information contained in the pages. A reader is an interpreter for a page (e.g., text formatters, graphic decoders, program interpreters, audio and video players, data manipulation systems); $M : S_S \rightarrow P$ is a *value function* mapping states into data objects (in the form of a page). S_S is the subset of S comprising basic states and OR states (AND states are not mapped into data objects); L is the visibility level used for defining the hierarchy depth when displaying pages during navigation; and $V : P \rightarrow R$ is the *visualization relationship* which associates each hyperdocument page with a single reader that is able to interpret it.

In short, HMBS uses the structure and execution semantics of statecharts [26] to specify the structural organization and the browsing semantics of hyperdocuments. Turine *et al.* use the term "browsing semantics" as referring to the dynamic properties of a reader's experience when navigating through a document, i.e., it is the manner in which information is to be visited and presented to the reader. In this sense, HBMS may be included in the category of behavioral models.

Besides that, HMBS is also adequate for describing the hierarchical structure common to many hyperdocuments since the hierarchy levels are directly mapped into the different levels of an underlying statechart model. In addition to the hierarchy mechanism, the model provides parallel and sequential decompositions with associated semantics.

Another major advantage of a statechart-based model is that it provides a mathematical model with associated semantics and algorithms, yet it has an easy and intuitive visual notation associated. It also has the advantage of being a very intuitive model, as statecharts are an extension of finite-state machines, and modeling based on states and events is well-established approach. Moreover, as statecharts were designed to model concurrent reactive systems, HMBS is suitable for describing concurrency aspects inherently associated to the navigation through a hyperdocument.

In the instructional level, we focused on the mechanisms for hierarchical decomposition HMBS provides, complementing the idea of hierarchical organization, already explored in the conceptual model. Observe that the notion of hierarchical decomposition can be related to the depth of the learning content to be presented.

As mentioned in Sect. 3.3, the establishment of *knowledge categories* is one of the modeling requirements related to the development of educational modules, being relevant to qualify the elements of the content in terms of their roles and instructional purposes. In order to make HMBS suitable for modeling such instructional aspects, it was extended for representing different knowledge categories. The idea was to allow the model representing information items (concepts, facts, principles and procedures) and instructional elements (explanatory, exploratory and evaluative).

As said before, the content of a hyperdocument in HMBS is represented by a finite set of information pages P. Each page $p \in P$ is conceptually defined by the triple $\langle c, t, Anc_p \rangle$, where c represents the information content, t is the title that uniquely identifies the page, and Anc_p refers to the collection of anchors contained in the page. The information content c can be expressed in any static (text, graphic or image) or dynamic (audio, video, animation) media. More information about the main features of HMBS and its formal definition can be found in [63].

To represent different knowledge categories into the HMBS model, we redefined the set of pages P to the triple $\langle C, t, Anc_p \rangle$, where C represents the finite set of information content, formalized by the quadruple $C = \langle I_{\text{Info}}, E_{\text{Explan}}, E_{\text{Explor}}, E_{\text{Eval}} \rangle$, where:

- I_{Info} represents the finite set of information items that constitute the content of the page. I_{Info} is defined by the quadruple $I_{\text{Info}} = \langle Conc, F, Princ, Proc \rangle$, where:
  - $Conc = \{conc_1, conc_2, \ldots, conc_i\}$, $i \geq 0$ is the finite set of concepts associated with the page;
  - $F = \{f_1, f_2, \ldots, f_j\}$, $j \geq 0$ is the finite set of facts associated with the page;
  - $Princ = \{princ_1, princ_2, \ldots, princ_k\}$, $k \geq 0$ is the finite set of principles associated with the page; and
  - $Proc = \{proc_1, proc_2, \ldots, proc_l\}$, $l \geq 0$ is the finite set of procedures associated with the page.
- E_{Explan} is the finite set of explanatory elements that constitute the content of the page. E_{Explan} is defined by the tuple $E_{\text{Explan}} = \langle Ex, Compl \rangle$, where:
  - $Ex = \{ex_1, ex_2, \ldots, ex_i\}$, $i \geq 0$ is the finite set of examples associated with the page; and
  - $Compl = \{compl_1, compl_2, \ldots, compl_j\}$, $j \geq 0$ is the finite set of complementary information associated with the page.
- E_{Explor} is the finite set of exploratory elements that constitute the content of the page. E_{Explor} is defined by $E_{\text{Explor}} = \langle Exer \rangle$, where:
  - $Exer = \{exer_1, exer_2, \ldots, exer_i\}$, $i \geq 0$ is the finite set of exercises associated with the page.
- E_{Eval} is the finite set of evaluative elements that constitute the content of the page. E_{Eval} is defined by the triple $E_{\text{Eval}} = \langle DE, FE, SE \rangle$, where:
  - $DE = \{de_1, de_2, \ldots, de_i\}$, $i \geq 0$ is the finite set of diagnostic evaluations associated with the page;
  - $FE = \{fe_1, fe_2, \ldots, fe_j\}$, $j \geq 0$ is the finite set of formative evaluations associated with the page; and
  - $SE = \{se_1, se_2, \ldots, se_k\}$, $k \geq 0$ is the finite set of summative evaluations associated with the page.

The extended version of HMBS for representing different knowledge categories is named *HMBS/Instructional* model.

4.3 Didactic model

As we said before, one of the main challenges in active learning scenarios is how to develop and deliver more flexible, adaptable and personalized educational products. The aim is that the user (learner or instructor) can be "free" to dynamically decide which topics to navigate, progressing more or less deeply into them. In other words, it is important to provide educational modules that can be traversed either in breadth or in depth, according to characteristics such as course length and type, instructor's preferences, learner's profiles and learning goals.

The *Didactic Model* is responsible for the establishment of precedence relations (prerequisites and sequences of presentation) among concepts, information items and instructional elements. In short, it can be used to illustrate the way the didactic space is modified while being navigated by the user, i.e., which information becomes active/inactive when a given path is traversed. Moreover, it is useful to represent dynamic contexts of learning, where the elements of the content are determined according to specific parameters defined in terms of the characteristics of the course, learners and instructors.

Since HMBS addresses relevant requirements under the didactic perspective (*history mechanisms*, *event propagation* and *learning contexts definition*), it was also adopted in order to construct the didactic model. Moreover, by using HMBS we can validate the learning content through the analysis of the subjacent statechart properties [63].

As an extension to HMBS at the didactic level, we introduced the idea of *open specifications*, providing support for the definition of dynamic contexts of learning. Depending on aspects such as audience, learning goals and course length, distinct ways for presenting and navigating through the same content can be required. An open specification allows representing all sequences of presentation in the same didactic model.

So, from a single model, several versions of the same content can be generated according to different pedagogical aspects. Additionally, when an educational module is implemented based on an open specification, its navigation paths can be defined by the user (the instructor, in case of traditional classes; the learner, in distance environments; or both, in case of blended learning), in "execution time". During the presentation, the user is able to dynamically decide which topics should be navigated and in which sequence based on the learner's skills, understanding and feedback, for instance.

Aiming at representing open specifications, we extended HMBS with the notion of *DD* (Dynamically Defined) *states*. In short, a *DD* state has the following properties:

1. Only one state can be active at a given time.
2. A *DD* state does not include the representation of initial state (default), which is dynamically defined by the user, in execution time.
3. All OR substates of a *DD* state (OR_{DD}) are totally connected to each other.
4. Hierarchy of *DD*-superstates—leaving a *DD* state X (X_{DD}) can activate the OR_{DD} states from the hierarchy of *DD*-superstates of X_{DD}.

Properties (1), (2) and (3) do not require any change in the HMBS syntax and semantics. Actually, the modifications are limited to the model's graphical notation—for the sake of legibility, transitions and events are implicitly represented. On the other hand, Property (4) requires the following extensions to the HMBS model (and to the subjacent statechart):

– Let the hierarchy of superstates of a given state X_{DD} be defined by the sequence:

$$SeqAncestors(X_{DD}) = ancestor_{S_1}, ancestor_{S_2}, ancestor_{S_3}, \ldots, ancestor_{S_i}$$

such that $ancestor_{S_{j-1}} \in \rho(ancestor_{S_j})$, $j = 1, \ldots, i-1$.

The hierarchy of *DD*-superstates of X_{DD} is defined by the sequence:

$$SeqAncestors_{DD}(X_{DD}) = ancestor_{S_1}, ancestor_{S_2}, ancestor_{S_3}, \ldots, ancestor_{S_n}$$

such that $\forall ancestor_{S_k}, \quad k = 1, \ldots, n$:

$\psi_{st}(ancestor_{S_k}) = OR_{DD}$

and $ancestor_{S_{k-1}} \in \rho(ancestor_{S_k})$,

where *Function Type* $\psi_{st} : S \to \{OR, OR_{DD}, AND\}$ defines the type of each *OR* state.

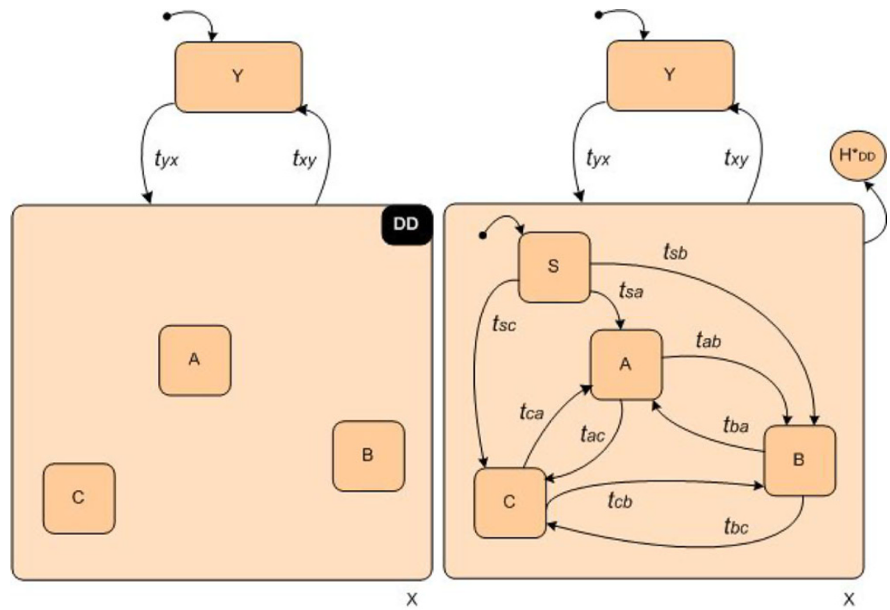
Both the notion of *DD* states as well as the hierarchy of *DD*-superstates help to establish open specifications since they allow to represent all sequences of presentation in the same didactic model.

Figure 4(a) illustrates the representation of a *DD* state X (X_{DD}). Figure 4(b) shows the internal behavior of X, explicitly representing its transitions and events. The notation H^*_{DD} represents the hierarchy of *DD*-superstates.

Notice that the representation of *DD* states (and open specifications) is related to the establishment of *learning contexts*. This modeling requirement, discussed in Sect. 3.3, is important to allow learners with different profiles and goals to explore the same material through distinct perspectives.

The extended version of HMBS to support *DD* states (and open specifications) is named *HMBS/Didactic*.

Fig. 4 Representation of a *DD* state



(a) *DD* state *X* (implicit transitions and events).

(b) Internal behavior of *X* (explicit transitions and events).

4.4 Guidelines for applying $\mathcal{IMA}$–$\mathcal{CID}$

Aiming at helping instructors and/or content developers on the application of $\mathcal{IMA}$–$\mathcal{CID}$, we have developed a set of 11 guidelines. Guidelines from (1) to (4) refer to the construction of the conceptual model; guidelines from (5) to (8) refer to the construction of the instructional model; guidelines (9) and (10) refer to the construction of the didactic model; and guideline (11) refers to the creation of the learning content according to the models.

1. *Identify the main concepts of the topic to be taught/learned.*
   - *Hint*: If you are using a source document (e.g., a book chapter, a technical report, a previous presentation, class notes) as the basis for creating the learning content, you can use the table of contents (or something similar that can provide the overall structure of the document) as a starting point to look for significant concepts.
   - *Hint*: Write down a list of main concepts. Inspect the source document looking for substantives used to describe the topic. They can help on identifying concepts.
2. *Identify taxonomy and composition relations among the concepts.*
   - *Hint*: Write down a list of concept taxonomy relations. Inspect the source document looking for verbs and expressions such as “is a type of” and “is classified in”. They can help on identifying concept taxonomy relations.
   - *Hint*: Write down a list of concept composition relations. Inspect the source document looking for verbs and expressions such as “is composed of”, “is formed by” and “is part of”. They can help on identifying concept composition relations.
3. *Identify domain-specific relations among the concepts.*
   - *Hint*: Write down a list of domain-specific relations. Inspect the source document looking for verbs and expressions such as “implies on”, “is a consequence of”, “assumes” and “uses”. They can help on identifying domain-specific relations.
4. *Construct the conceptual model, graphically representing the identified concepts and the relations among them.*
   - *Hint*: Check and revise all the concepts and relations you have identified. If necessary, return to the previous steps in order to include/exclude any concept and/or relation.
5. *For each concept represented in the conceptual model, identify its definition and the related information items.*
   - *Hint*: Inspect the source document looking for concept definitions. Mark in which parts of the document they can be found.
   - *Hint*: Inspect the source document looking for facts, procedures and principles that can be related to each concept. Mark in which parts of the document they can be found.

6. *For each concept and information item, identify the related instructional elements.*

   – *Hint*: Inspect the source document looking for examples, hints, suggestions of study, exercises and evaluations that can be related to each concept and information item. Mark in which parts of the document they can be found.

7. *For each concept, information item and instructional element, specify the media to be used.*

   – *Hint*: Inspect the source document looking for discrete (text, code, graphic, link, figure) and continuous (animation, video, audio, simulation) media that can be related to concepts, information items and instructional elements. Mark in which parts of the document they can be found.

8. *Construct the instructional model, graphically representing the concepts, information items, instructional elements and media previously identified by means of a statechart.*

   – *Hint*: Information items related to the same concept are generally represented as orthogonal (AND) states.
   – *Hint*: Concept taxonomy and composition relations generally suggest a representation by means of hierarchical decompositions and OR states.
   – *Hint*: Domain-specific relations generally suggest a representation by means of OR states.
   – *Hint*: Instructional elements are related to a concept or an information item by means of explicit transitions among states.
   – *Hint*: Check and revise all the elements you have identified. If necessary, return to the previous steps in order to include/exclude any element.

9. *Identify the precedence relations*[1] *among the elements represented in the instructional model.*

   – *Hint*: Write down a list of precedence relations. Consider characteristics such as course length and type (traditional, distance, blended), instructor's preferences, learner's profiles and cognitive styles, learning goals, among others.

10. *Construct the didactic model, providing a sequence to present the concepts, information items and instructional elements.*

   – *Hint*: If the didactic model establishes an open specification, there is no precedence relation to be represented. In this case, all relations among the elements should be represented by means of implicit transitions (DD states and hierarchy of DD-superstates).
   – *Hint*: If the didactic model establishes a close specification, precedence relations suggest a representation by means of explicit transitions.
   – *Hint*: If the didactic model establishes a partially open specification, precedence relations suggest a representation by means of explicit transitions and all other relations should be represented by means of implicit transitions (DD states and hierarchy of DD-superstates).
   – *Hint*: Check and revise all the relations you have identified. If necessary, return to the previous steps in order to include/exclude any relation.

11. *Create the learning content according to the $\mathcal{IMA}$–$\mathcal{CID}$ models developed.*

   – *Hint*: Each state in the didactic model can correspond to one or more slides (or pages) in the learning material, depending on the amount of information to be presented.
   – *Hint*: Hierarchical decomposition suggests a representation by means of a navigation menu.
   – *Hint*: Both implicit and explicit transitions among states suggest a representation by means of links in the navigation menu.
   – *Hint*: Relations among concepts/information items and instructional elements suggest a representation by means of buttons.
   – *Hint*: Besides the main slides (or pages), some specific slides (or pages) regarding references and summary can be created. Links for complementary materials, such as websites, working documents and tools, can be added as well.

In this section we summarized the main points of the $\mathcal{IMA}$–$\mathcal{CID}$ application by means of a set of guidelines. It is important to highlight that everyone interested in teaching and learning can benefit from $\mathcal{IMA}$–$\mathcal{CID}$: instructors, domain experts, content developers, educators and training professionals, among others.

On the one hand, $\mathcal{IMA}$–$\mathcal{CID}$ can be used not only to help the development "from scratch" of learning materials, either individually or collaboratively, but also to evolve, reuse and even re-engineer the existing materials. Furthermore, it is always possible to return for reviewing and revising the models and, as a consequence, the materials under construction. In a related perspective, $\mathcal{IMA}$–$\mathcal{CID}$ can also help on detecting faults and omissions in the source documents. For instance, if a concept definition is missing, such an omission can be detected by using the model-

[1]Notice that precedence relations are only needed for close and partially open specifications. For open specifications, all elements are totally connected to each other by means of implicit transitions. In this case, Guideline 9 can be skipped.

ing approach. Finally, if the material is systematically well-designed and developed, it may be easily evolved to be adequate to curriculum evolution.

On the other hand, the main concern for the $\mathcal{IMA}$–$\mathcal{CID}$ application is the need to be familiar with the structure and execution semantics of statecharts, what can lead to increasing costs to initially develop the materials based on the ideas discussed herein. However, there is often a learning curve that should be addressed in any method or model proposed. For instance, it can be not so intuitive for a non-specialist to address the learning curve associated with the adoption of SCORM, IMS CP, IMS LD (or any other educational method or model). Besides that, despite the costs, the quality factors of the $\mathcal{IMA}$–$\mathcal{CID}$-based modules, such as evolvability, maintainability and reusability, would increase the long-term benefits and decrease the overall costs.

As a final remark, it is also important to observe that even working in a lower, fine-grained level of abstraction (modeling each single concept, information item and their relationships), $\mathcal{IMA}$–$\mathcal{CID}$ can be used in a complementary way with models working in higher, coordination levels of abstraction. For the sake of illustration, suppose that an $\mathcal{IMA}$–$\mathcal{CID}$-based content is exported as, for instance, a SCORM package (in the same way it happens for any other content). After this process, the $\mathcal{IMA}$–$\mathcal{CID}$-based content could be: (1) described by metadata; (2) organized as a structured collection of one or more learning objects; (3) packaged in such a way that it could be imported, delivered and tracked by a SCORM-compliant learning management system; and (4) truly portable, i.e., it could be delivered by any web server without additional special server-side components or installation. Notice that $\mathcal{IMA}$–$\mathcal{CID}$ did not focus on issues from (1) to (4) since they have already been addressed by SCORM. We intend to better investigate such complementary perspective in short-term, as a further work.

$\mathcal{IMA}$–$\mathcal{CID}$ has been applied into the development of educational modules for different domains: software testing, code inspection, programming foundations, open source methods and technologies, critical embedded systems, and elementary materials on mathematics. For the sake of illustration, next we discuss the $\mathcal{IMA}$–$\mathcal{CID}$ application for the software testing domain.

5 An example of the $\mathcal{IMA}$–$\mathcal{CID}$ application

The $\mathcal{IMA}$–$\mathcal{CID}$ approach was applied as part of the development process of SOFTTEST—an educational module for the software testing domain. We chose the testing area since it is one of the most relevant activities regarding software development but, at the same time, it is a difficult topic to learn or teach without the appropriate supporting mechanisms [60].

Considering this scenario, the learning goal was to foster theoretical, empirical and tool specific knowledge by providing learners with a broad and deep view of the testing activity fundamentals and of their practical application by mastering testing tools.

The module was composed of 16 sub-modules, addressing relevant topics of software testing such as testing fundamentals, testing phases, testing techniques and criteria, theoretical and empirical studies, among others. Besides that, some sub-modules were specifically designed for motivating, illustrating and practising the concepts addressed in the other sub-modules. Also, specific testing tools were integrated to the module, acting as supporting mechanisms for conducting the proposed instructional activities. The whole structure of SOFTTEST was graphically represented in terms of a conceptual map. For the sake of space, this structure is not discussed here. Details can be found in [5].

For each sub-module, concepts, facts, principles, procedures, examples and exercises were modeled and implemented as a set of slides, integrated to HTML pages, text documents, learning environments and testing tools. Figure 5 shows an overview of SOFTTEST,[2] presenting its main components and their integration.

The $\mathcal{IMA}$–$\mathcal{CID}$ models were developed for each one of the 16 sub-modules of SOFTTEST, according to the guidelines discussed in Sect. 4.4. Next we illustrate the construction of the $\mathcal{IMA}$–$\mathcal{CID}$ models for a particular subject of the Testing Techniques sub-module—the mutation analysis testing criterion [15].

5.1 The $\mathcal{IMA}$–$\mathcal{CID}$ models for mutation analysis

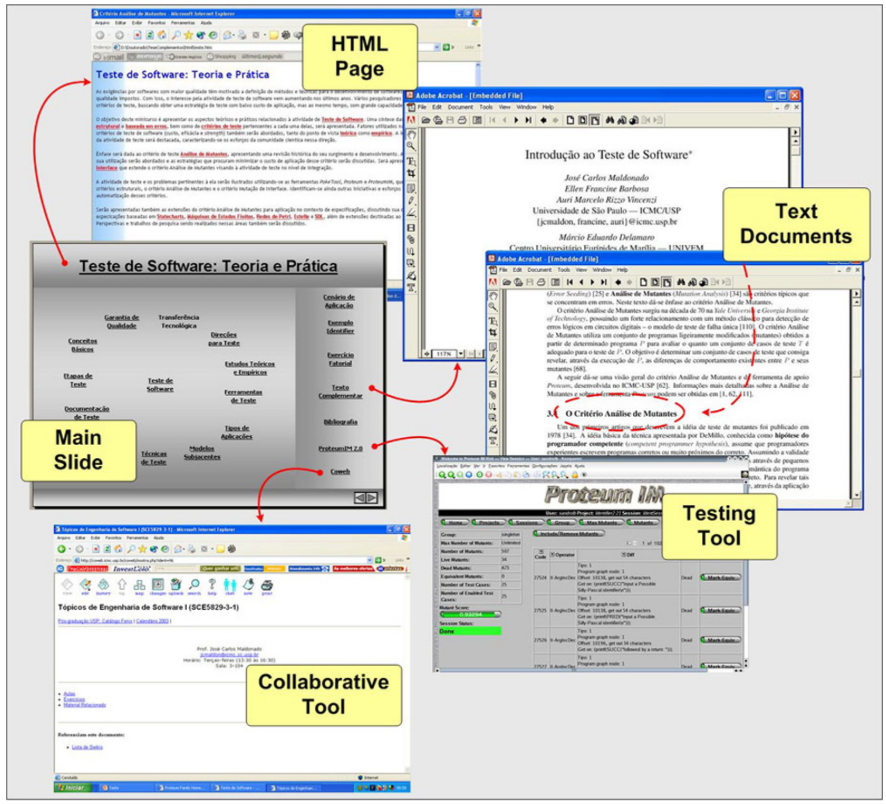
Mutation analysis is an error-based testing criterion [15]. In short, simple faults are introduced into a program P under testing by creating different versions of P, known as mutants, each of which containing a simple syntactic change. The quality of a test set T is measured by its ability to distinguish the behavior of the mutants from the behavior of the original program. If P behaves as per the specification when T is applied, then the quality of P is demonstrated; otherwise, a fault has been detected and the debugging activity will take place.

For constructing the conceptual model, $\mathcal{IMA}$–$\mathcal{CID}$ follows the rules of conceptual mapping [48], also including particular notations to represent specific types of relationship: concept taxonomy (*type-of*) and concept composition (*part-of*).

Figure 6 shows the conceptual model developed for mutation analysis. From this model, we can infer that: (1) mutation analysis is one of the testing criteria of the error-based technique; (2) mutation analysis assumes the principles of competent programmer hypothesis and coupling

[2] SOFTTEST was developed in Portuguese.

Fig. 5 SOFTTEST: an educational module for software testing



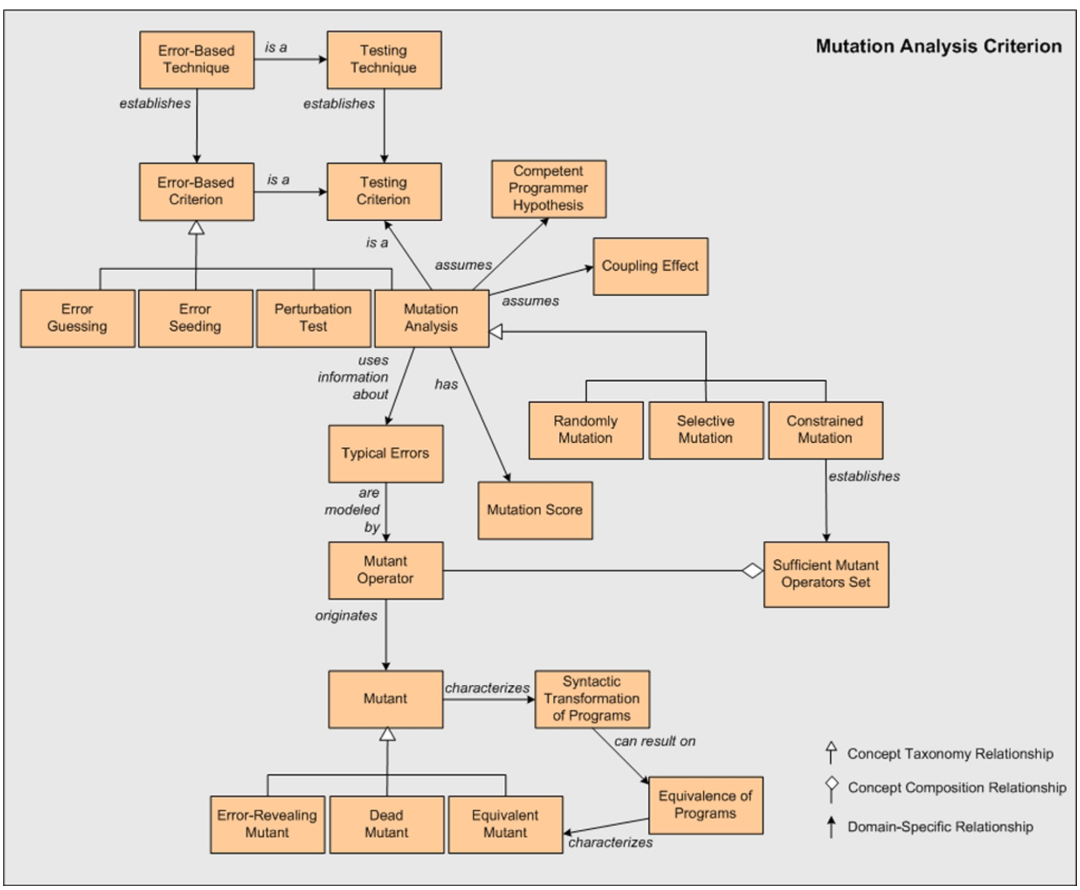


Fig. 6 Conceptual model for mutation analysis

effect; (3) mutation analysis uses information about typical errors, which are modeled by mutant operators in order to generate the mutant programs; (4) mutation analysis has a mutation score associated with; (5) mutants characterize syntactic transformations of programs and can be classified as error-revealing, dead or equivalent ones; (6) syntactic transformation of programs can result on equivalence of programs, characterizing an equivalent mutant; (7) randomly mutation, selective mutation and constrained mutation are variants of mutation analysis; and (8) constrained mutation can be used in the establishment of sufficient mutant operators sets, which represent subsets of mutant operators.

Notice that concept taxonomy relationships were used for representing (1), (5) and (7), while a concept composition relationship was used for representing (8). Domain-specific relations were adopted in the other cases.

It is also important to point out the subjective aspect as it relates to the conceptual modeling activity. In fact, different conceptual models can be developed for the same knowledge, depending on the domain expert and/or designer responsible for its construction.

Regarding the construction of the instructional model, it consists of specifying other types of information that can be associated to the concepts previously identified. Basically, information items and instructional elements are represented in the model. In the context of mutation analysis, information items could be, for instance:

– Concepts' definition: definitions of mutant (dead, equivalent, error-revealing), mutant operator, mutation score, and so on;
– Facts: when mutation analysis was proposed and by whom;
– Principles: competent programmer hypothesis and coupling effect; and
– Procedures: basic steps for applying mutation analysis.

With respect to the instructional elements, examples could be excerpts of mutants generated by specific operators, while exercises and evaluations could be proposed in terms of tasks involving the application of mutation analysis through automated tools.

Figure 7 shows the instructional model, constructed according to the *HMBS/Instructional*, for the mutation analysis criterion. First, we need to specify the information items. Consider, for instance, the `MutationAnalysis` state. In addition to the concept definition (`MA:concept:text`), some related facts (`MA:fact:text`) and principles (`CompetentProgrammer:principle:text` and `CouplingEffect:principle:text`) are specified. Notice that the media related to each information item is also specified. For instance, consider the `Application` state within the `MutationAnalysisDetails` state. A procedure is specified in terms of textual (`ApplicationMA:procedure:text`) and graphical (`ApplicationMA:procedure:figure`) representations. Figure 8 illustrates how the different knowledge categories of the `Application` state were implemented in the educational module. Similarly, other concepts and information items related to the mutation analysis domain were modeled into separate states: `MutantOperator`, `MutantGeneral`, `MutationScore`, `ApproachesGeneral`, and so on.

Next, we have to determine the instructional elements. In case of SOFTTEST, only explanatory and exploratory elements were considered. As explanatory elements take, for instance, the states `Mutant:example:figure`, `DeadMutant:example:figure`, `EquivalentMutant:example:figure` and `ErrorRevealingMutant:example:figure`. These states represent examples related to the concepts of a mutant program (`Mut:concept:text`), a dead mutant (`DeadMutant:concept:text`), an equivalent mutant (`EquivalentMutant:concept:text`) and an error-revealing mutant (`ErrorRevealing Mutant:concept:text`), respectively. Notice that these examples are also related to another explanatory element (`IdentifierImplementation:complementary:figure`), which provides complementary information for them.

The exercise represented by the state `ApplicationMA:exercise:text` corresponds to the exploratory element for mutation analysis. Basically, it consists of the application of the mutation analysis criterion to test a given program (in our module, the factorial one). Explanatory elements, included to provide some help for solving the exercise, are represented by the states `FactorialImplementation:complementary:figure` and `FactorialHintMA:complementary:figure`. Notice that the required tools for performing the exercise are also modeled. The `Coweb:tool` state represents a collaborative learning environment (*CoWeb* [18]), used as a discussion space among learners and instructors. The `ProteumIM:tool` state corresponds to a specific testing tool (*Proteum* [14]), used for applying the mutation analysis criterion. Figure 9 illustrates the proposed exercise regarding the mutation analysis application.

The last model, the didactic one, consists of defining the sequences for presenting all the components of the knowledge domain. Figure 10 illustrates part of the didactic model,[3] constructed according to the *HMBS/Didactic*, for the mutation analysis criterion. It corresponds to an *open specification*, in which all possible sequences of presentation among the modeled objects are represented. Consider, for instance, the `MutationAnalysisDetails` state. By exploring the notion of *DD* states, the `MutationAnalysisDetails` substates (OR_{DD} states)—`MutantOperator`,

[3]For the sake of space, explanatory and exploratory elements were not illustrated.

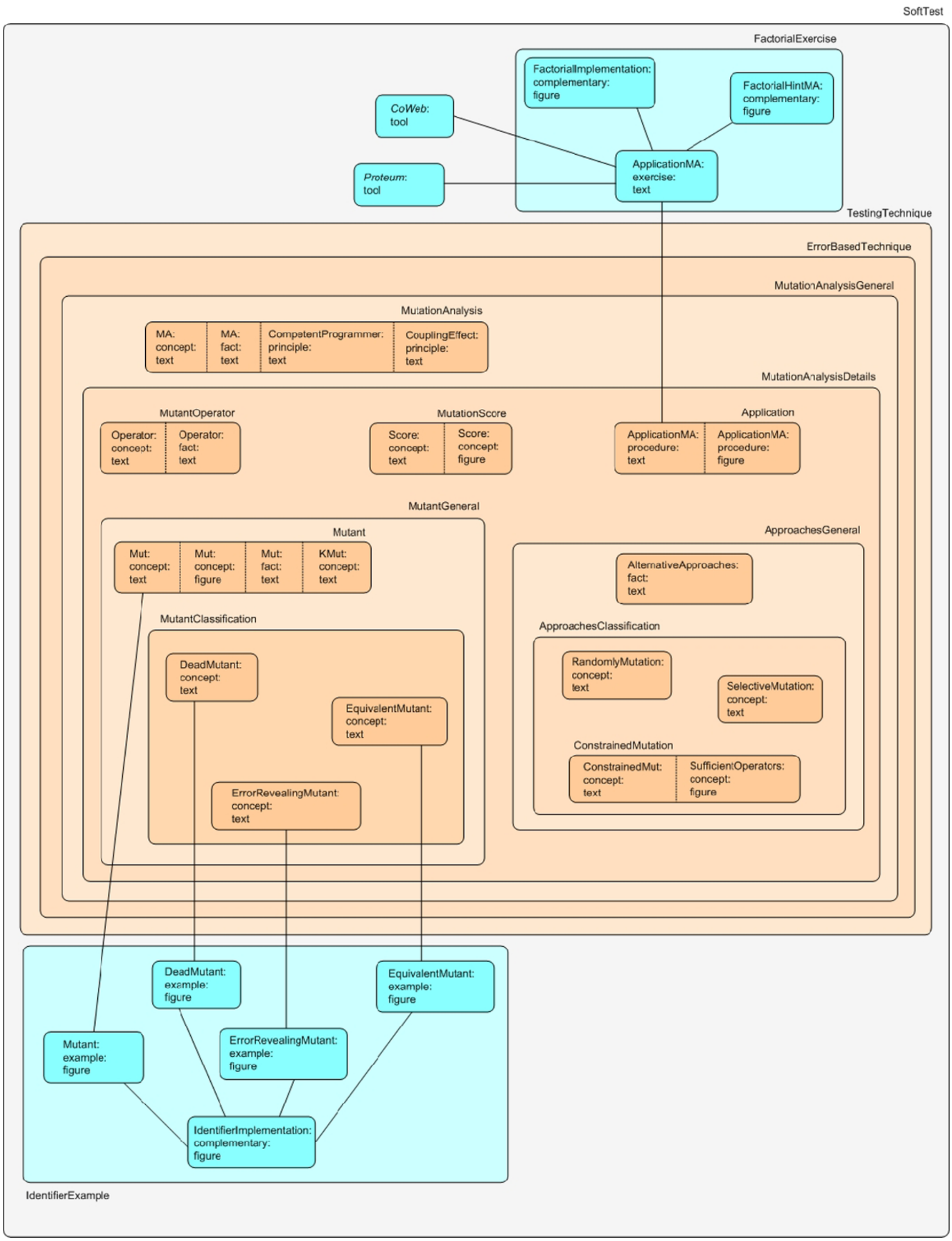


Fig. 7 Instructional model for mutation analysis

`MutantGeneral`, `MutationScore`, `Application` and `ApproachesGeneral`—are all connected to each other by implicit transitions, which are responsible for establishing the navigation paths among them. So, from `MutantOperator` we can get to the states `MutantGeneral`, `MutationScore`, `Application` and `ApproachesGeneral` (and vice versa). Similarly, consider the `Mutant` state. From `Mutant` we are able to get to `MutantClassification`

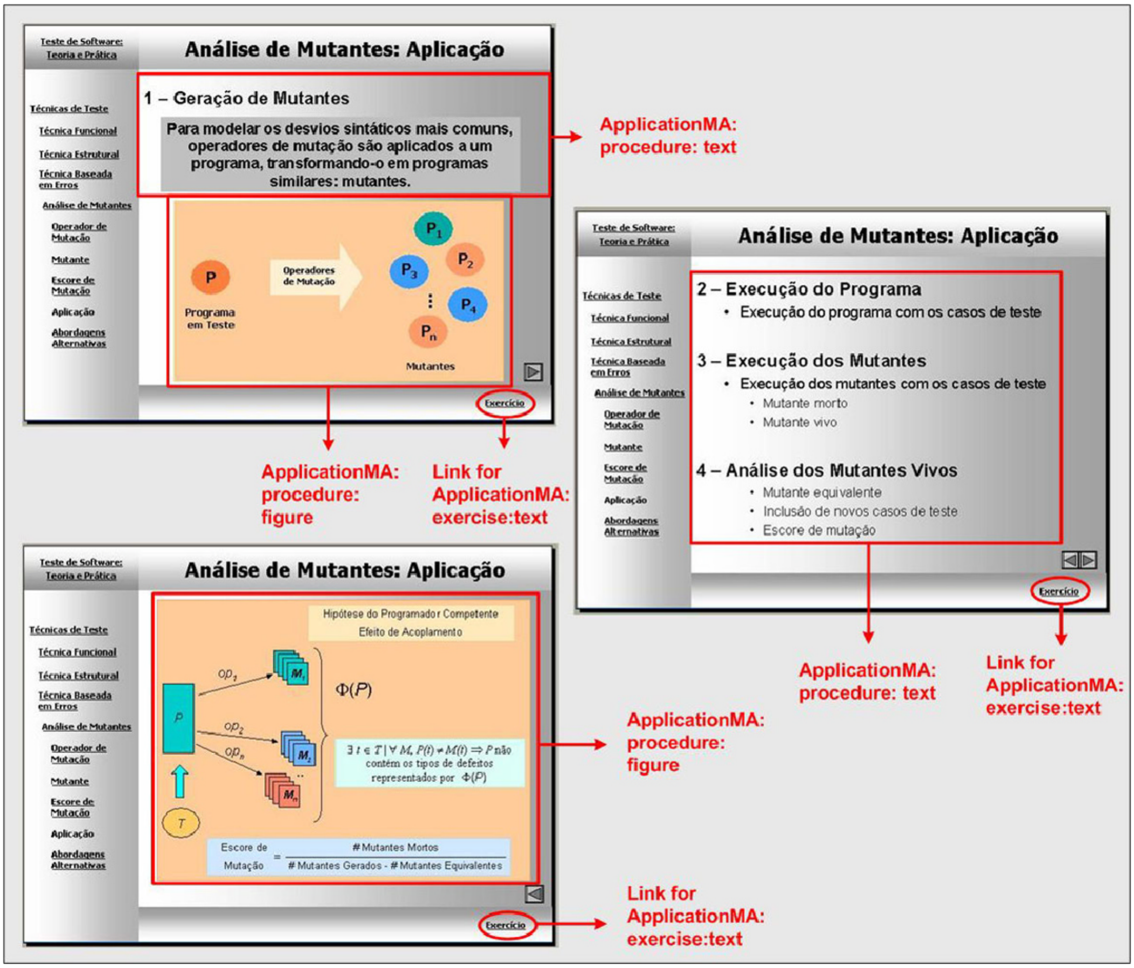


Fig. 8 Slides implementing the different knowledge categories of the `Application` state

(and vice versa). Actually, both states are substates of `MutantGeneral` (*DD* state) and, in this sense, they are connected to each other by means of implicit transitions.

We can also explore the idea of an hierarchy of *DD*-superstates. For instance, consider the sequence (`MutantGeneral`, `MutationAnalysisDetails`, `MutationAnalysisGeneral`, `ErrorBasedTechnique` and `TestingTechnique`) as the hierarchy of *DD*-superstates of the `Mutant` state. According to this hierarchy, from `Mutant` we can reach all OR_{DD} states of `MutationAnalysisDetails`. To define the full set of states we can reach from `Mutant`, the same analysis should be carried out for all states of the hierarchy of *DD*-superstates of `Mutant`. Notice that we cannot get to the states `AlternativeApproaches` and `ApproachesClassification` from the `Mutant` state since `ApproachesGeneral` does not pertain to the hierarchy of *DD*-superstates of `Mutant`.

Additionally to the definition of an *open specification*, definitions of a *partially open specification* and of a *close specification* can also be established in the scope of the didactic model. Basically, in a partially open specification, while some sequences of presentation can be established in "execution time", others are previously defined by the domain expert and/or the instructor during the development of the module. Indeed, instead of having just implicit transitions, the idea is to make some of them be explicitly represented in the didactic model. On the other hand, in a close specification all sequences are predefined, i.e., only one fixed sequence of presentation is available in the module. In this case, the transitions are explicitly represented.

Notice that the sequences of presentation derived from partially open specifications and from close specifications represent subsets of the total set of sequences established by an open specification. As highlighted before, a didactic model defined in terms of an open specification can be seen as the basis from which all sequences of presentation are derived. So, by using the didactic model illustrated in Fig. 10, several implementations of the same content about mutation analysis can be obtained. Such characteristic is essential to

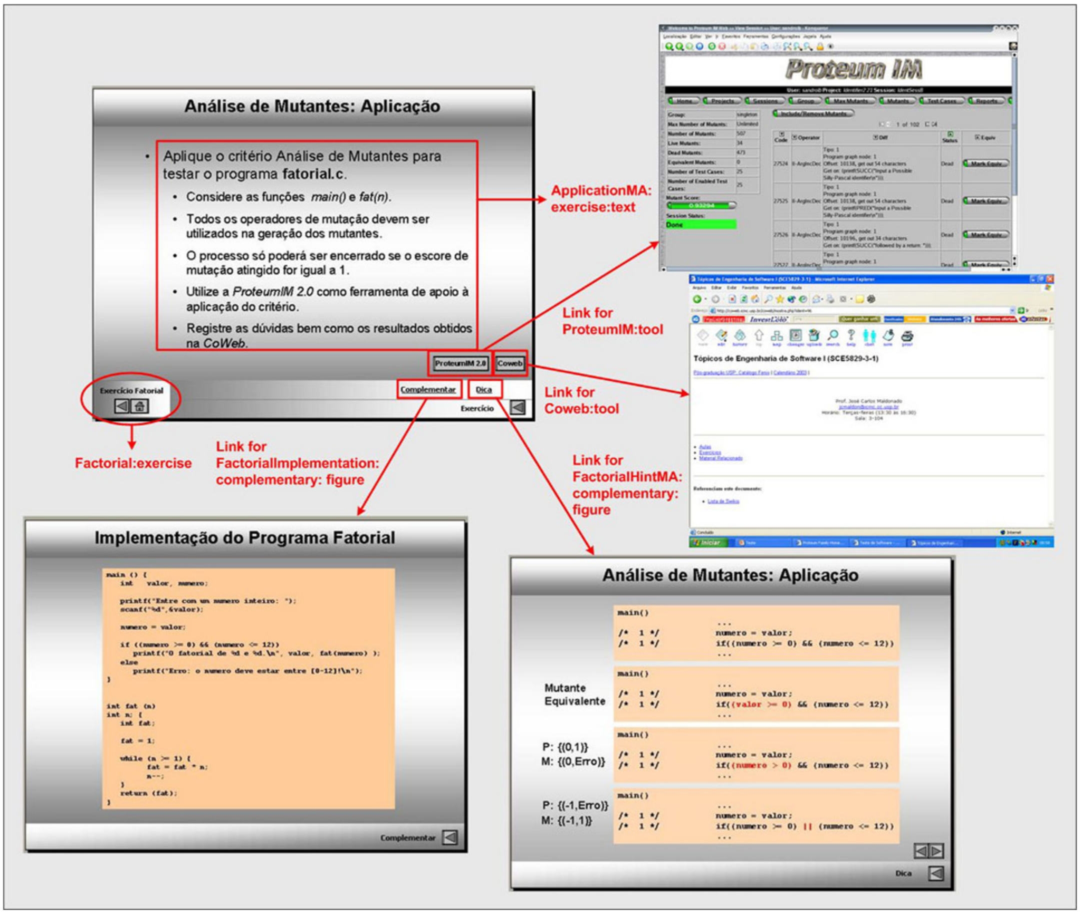


Fig. 9 Slides implementing the proposed exercise for mutation analysis application

generate differentiated content, whose topics, depth and sequences of presentation are established according to some particular aspects (e.g. course length, learning goals, instructor's preference, learner's profile).

Figure 11 illustrates part of the didactic model for mutation analysis constructed by using the idea of a close specification. Notice that all sequences of presentation are predefined. In fact, none of the states represented in the model was specified as a *DD*-state. Also, all possible transitions between states were explicitly represented. Again, consider the `Mutant` state. Now, from `Mutant` we are able to get only two states: `DeadMutant` and `MutationScore`.

The decision on which kind of specification to use should be based on the users as well as on the expected characteristics of the module. For instance, one strength of open specifications is the flexibility to navigate through the material according to the feedback and questions of the audience. On the other side, the instructor has to make sure to achieve the objectives of the lessons in order to keep the learners localized. Thus, while for less experienced instructors a close specification seems to be the better choice, for the most experienced ones, an open specification would be an adequate alternative too.

Figure 12 shows an overview of the resulting mutation analysis material, modeled according to $\mathcal{IMA}$–$\mathcal{CID}$ as part of the SOFTTEST educational module.

6 $\mathcal{IMA}$–$\mathcal{CID}$ application: lessons learned

In the previous section we have focused on the $\mathcal{IMA}$–$\mathcal{CID}$ application in the development of an educational module for teaching software testing. However, besides software testing, $\mathcal{IMA}$–$\mathcal{CID}$ has been applied into the development of educational modules for different domains, such as code inspection, programming foundations, open source methods and technologies, computer networks and elementary materials on mathematics. Particularly, the materials produced in the context of two broad projects have been developed according to the modeling approach: (1) *QualiPSo* (Quality Platform for Open Source Software),[4] funded by the European Community; and (2) *INCT-SEC* (National Institute of

[4] www.qualipso.org.

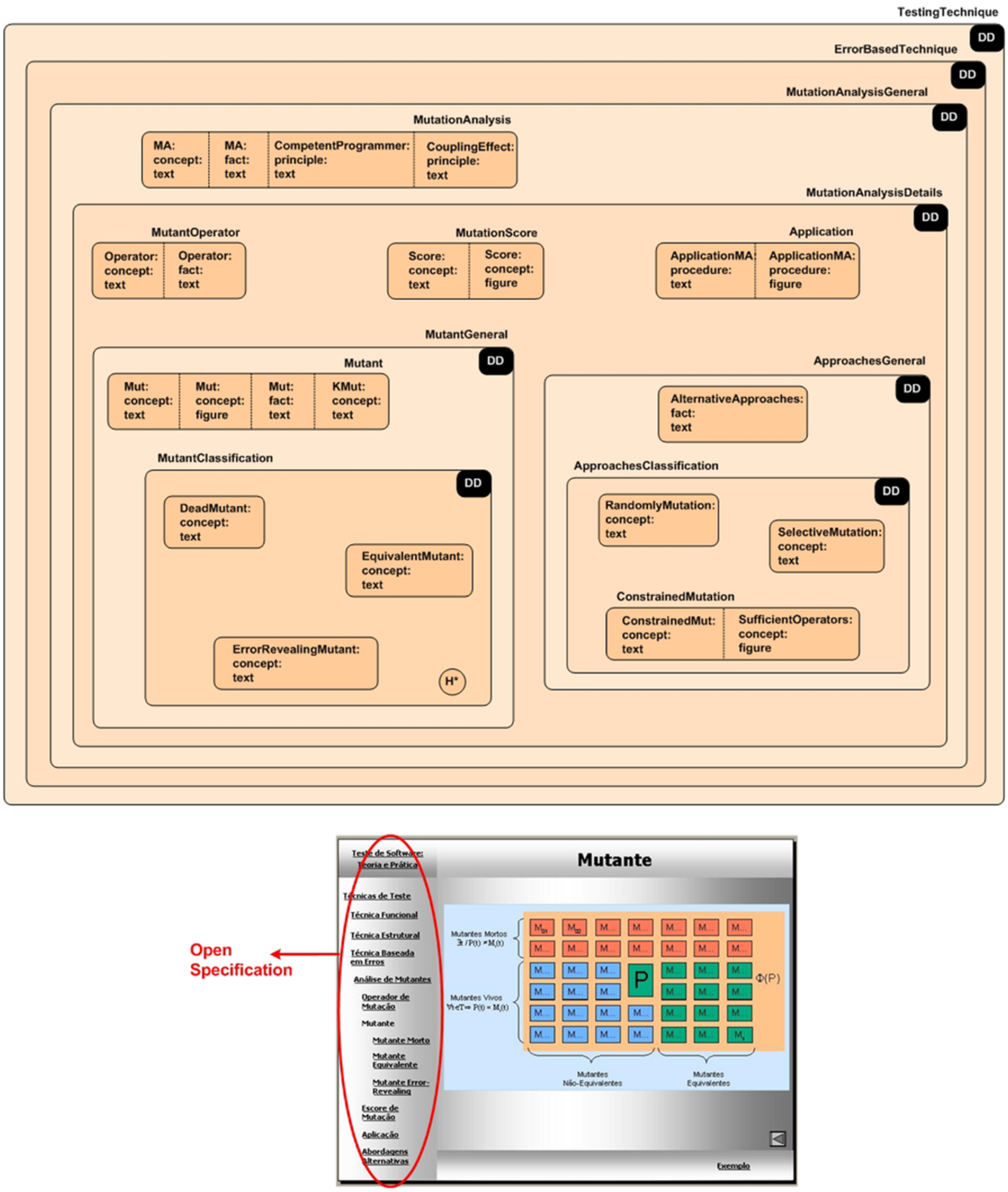


Fig. 10 Didactic model/slides for mutation analysis: open specification

Science and Technology—Critical Embedded Systems),[5] financed by the Brazilian funding agencies.

Next we discuss the application of $\mathcal{IMA}$–$\mathcal{CID}$ considering both the author's as well as the learner's perspectives on using the approach and its resulting educational modules are considered. Firstly, we provide an overview of the main modules produced according to $\mathcal{IMA}$–$\mathcal{CID}$ summarizing the authoring data collected during their development. Secondly, we focus on the learner's attitude toward using the SOFTTEST educational module. Finally, we evaluate the learning effectiveness provided by an $\mathcal{IMA}$–$\mathcal{CID}$based educational module for teaching software testing and code inspection activities (ITONCODE).

[5] www.inct-sec.org.

6.1 An overview of the $\mathcal{IMA}$–$\mathcal{CID}$-based educational modules

In this section we discuss the application of $\mathcal{IMA}$–$\mathcal{CID}$ in the development of educational modules for different knowledge domains, focusing on the authors' attitudes toward using the modeling approach.

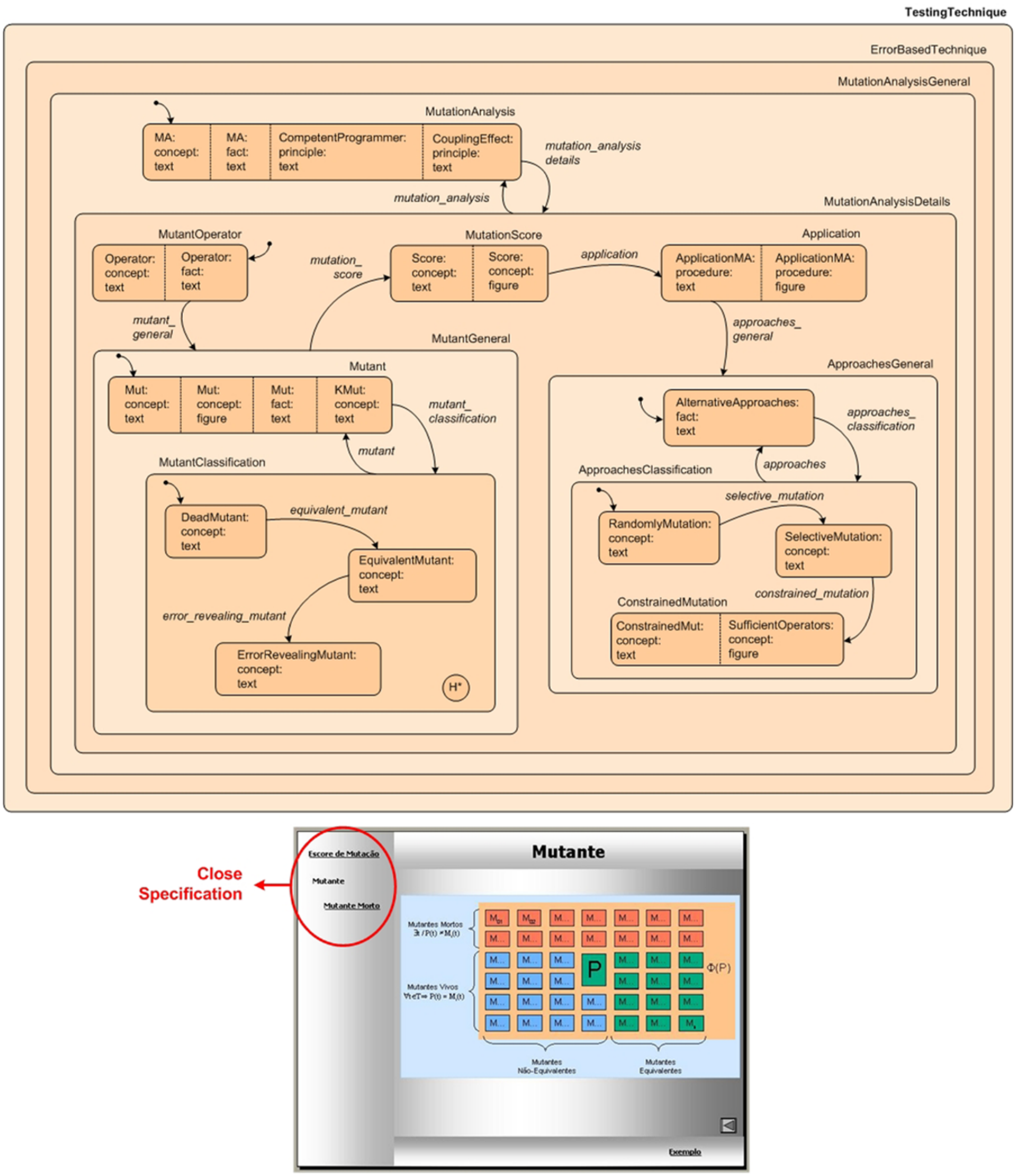


Fig. 11 Didactic model/slides for mutation analysis: close specification

Table 2 summarizes the main characteristics of eight $\mathcal{IMA}$–$\mathcal{CID}$-based educational modules. The first modules—SOFTTEST, ITONCODE and JABUTI-TT—are related to verification and validation (V&V) activities, more specifically to software testing, code inspection and testing tools.

SM-VTM is in the context experimental software engineering, addressing issues of systematic mapping and visual text mining. TECHINT and OMM are in the context of the *QualiPSo* project. The first deals with issues of technical interoperability for open source systems while the last is related to the proposal of an open maturity model. NUM-RAC deals with rational numbers in elementary teaching of mathematics; and COMPNET is in the context of the *INCT-SEC* project, dealing with the teaching of computer networks.

The target audience was diversified, varying from high school students to grad/undergrad students, professionals from industry and project members. Most modules were developed to be used in face-to-face, blended or distance learning courses. Exceptions are ITONCODE, SM-VTM and NUM-RAC, which were specifically designed for face-to-face courses.

Table 3 provides some authoring information regarding the produced modules. It is important to highlight that the

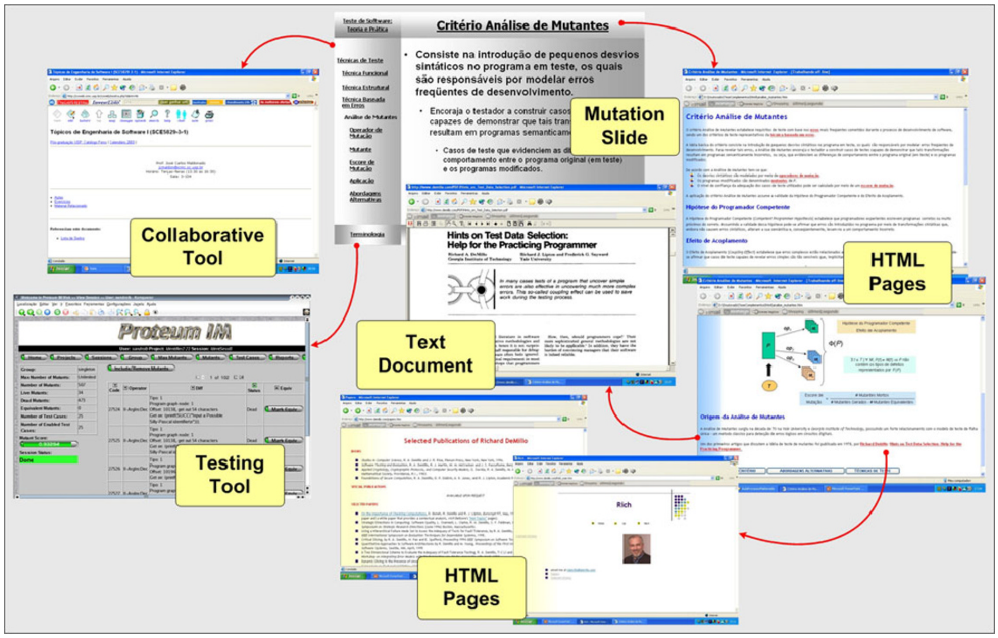


Fig. 12 Main components of the learning material for mutation analysis

Table 2 $\mathcal{IMA}$–$\mathcal{CID}$-based educational modules: an overview

Educational module	Subject	Course type	Target audience
SoftTest	Software Testing	Short-course	Undergrad and grad students, SE practitioners
ITonCode	Software Testing and Code Inspection	Undergraduate course	Undergrad students
JaBUTi-TT	JaBUTi Testing Tool	Short-course	Undergrad and grad students, SE practitioners
SM-VTM	Systematic Mapping and Visual Text Mining	Conference presentation	Grad students, professionals from industry
TechInt	Open Source Technologies	Short-course	QualiPSo members and users
OMM	Open Source Methods	Short-course	QualiPSo members and users
Num-Rac	Rational Numbers	Complementary course	High school students on mathematics
CompNet	Computer Networks	Short-course	Undergrad and grad students, network administrators and security professionals, INCT-SEC members and users

Table 3 $\mathcal{IMA}$–$\mathcal{CID}$-based educational modules: authoring information

Educational module	Author expertise	Dev. type	Effort (man-month)	Concepts	II	IE	Presentation mechanism
SoftTest	Domain/IMA-CID	Re-engineering	2	110	93	24	258 slides
ITonCode	Domain/IMA-CID	Re-engineering	1	213	207	74	423 slides
JaBUTi-TT	Domain/IMA-CID	Re-engineering	1.5	182	187	41	301 slides
SM-VTM	Domain	From scratch	0.25	23	20	5	96 slides
TechInt	IMA-CID	From scratch	0.25	21	63	15	116 slides
OMM	Domain	From scratch	0.5	15	20	8	99 slides
Num-Rac	Domain/IMA-CID	From scratch	Under dev.	13	18	9	Flash file
CompNet	IMA-CID	Re-engineering	Under dev.	27	31	1	–

authors' expertise was also diversified: some authors were domain experts; others were $\mathcal{IMA}$–$\mathcal{CID}$ specialists; and others knew both the approach as well as the subject domain. Furthermore, some modules were developed from scratch, while others were re-engineered from existing materials.

SOFTTEST, described in the previous section, was developed with basis on the knowledge managed by a software testing expert as well as on previous presentations and working documents related to the subject domain. In the end of the re-engineering process, 110 concepts, 93 information items (II) and 24 instructional elements (IE) were identified, resulting on an educational module with 258 slides. The development team was composed of three members: (1) a teacher, acting as the domain expert as well as the instructor of the module; (2) a graduate student, performing the roles of project manager, version manager, coordinator and developer; and (3) an undergraduate student, acting as developer and technician. All members were $\mathcal{IMA}$–$\mathcal{CID}$ specialists and worked collaboratively for developing the module.

ITONCODE, an educational module for teaching code inspection and software testing, was re-engineered from a lab package training material on V&V [7, 38]. Inspection/testing specialists and $\mathcal{IMA}$–$\mathcal{CID}$ experts were involved in the re-engineering process. Shortly, they merged the re-engineered material regarding code inspection with the software testing material alrealdy produced. Indeed, the part of ITONCODE addressing software testing issues corresponds to the SOFTTEST module. In the end of the re-engineering process, 213 concepts, 207 information items and 74 instructional elements were identified, and 423 slides were produced according to the $\mathcal{IMA}$–$\mathcal{CID}$ approach. The application of ITONCODE in a real learning scenario will be discussed in Sect. 6.3.

JABUTI-TT was an educational module developed for teaching the main concepts of *JaBUTi* (Java Bytecode Understand and Testing) [67]—an automated tool for the structural testing of Java programs/components. Part of the module was based on SOFTTEST, including specific concepts, information items and instructional elements regarding the testing tool. Similarly to SOFTTEST and ITONCODE, JABUTI was developed by testing specialists and $\mathcal{IMA}$–$\mathcal{CID}$ experts as well.

SM-VTM is an interesting application of $\mathcal{IMA}$–$\mathcal{CID}$ approach. The module was developed from scratch and the author had no previous knowledge about $\mathcal{IMA}$–$\mathcal{CID}$. The approach was applied according to the guidelines and no help was required from an $\mathcal{IMA}$–$\mathcal{CID}$ expert. As the module was referred to a conference presentation, emphasis was given to concepts (23) and information items (20). We received an informal, but very positive feedback from the author, especially with respect to the benefits of having a learning content well-structured and to the several different possibilities of navigation through the same content.

TECHINTER is another interesting application. It was developed from scratch, but differently from SM-VTM, the author was an $\mathcal{IMA}$–$\mathcal{CID}$ expert with no knowledge on the subject domain. The module was developed based on a set of textual documents and, for this reason, several information items (63), especially *facts*, were identified. Before delivery, the module was validated and approved by domain specialists. As highlighted by the author, additionally to the development of TECHINTER, the source documents could also be evolved based on the faults and omissions detected when applying $\mathcal{IMA}$–$\mathcal{CID}$.

OMM was also developed from scratch, by a domain expert with no knowledge on $\mathcal{IMA}$–$\mathcal{CID}$ application. 96 slides were developed according to the $\mathcal{IMA}$–$\mathcal{CID}$ models: 15 concepts were considered as part of the conceptual model, and 20 information items and 8 instructional elements were identified to compose the instructional and didactic models.

NUM-RAC is a differentiated module since it deals with the use of technology in engaging ways to help high school learners on developing mathematical skills. Furthermore, the module consists of an animated Flash file instead of a set of slides. NUM-RAC has been developed from scratch, by domain specialists together with $\mathcal{IMA}$–$\mathcal{CID}$ experts, and few specialized intervention has been required so far.

The last module, COMPNET, is also under development. The module is part of a course on Network Security, previously developed by researchers, professors and domain experts, delivered in the scope of the *INCT-SEC* project. The entire course on Network Security is organized into 20 classes, each of them having around 50 slides. Basically, the development consists of the re-engineering of the original course to the new format. The original set of slides has been analyzed by an $\mathcal{IMA}$–$\mathcal{CID}$ expert and the related $\mathcal{IMA}$–$\mathcal{CID}$ models have been created. Based on such models, a new set of slides will be created according to the approach. In the end of the re-engineering process, each class will correspond to an educational module.

From the authors' point of view, one benefit observed by applying $\mathcal{IMA}$–$\mathcal{CID}$ in the development of the educational modules was the resulting available documentation, mainly in terms of the produced models (conceptual, instructional and didactic). Besides helping to structure and organize the concepts and related information, such models were used as the instructional design rationale, playing a key role to easier evolve and maintain the modules after the delivery.

$\mathcal{IMA}$–$\mathcal{CID}$ was also useful to help on detecting faults and omissions during the re-engineering process. Thus, if a concept definition is missing in the source document, such an omission can be more easily detected by constructing the instructional model. The authors also pointed out as a key characteristic of $\mathcal{IMA}$–$\mathcal{CID}$ the possibility to always return for reviewing and revising the models, contributing for increasing the quality of the materials being produced.

Other significant result observed was the flexibility provided by the $\mathcal{IMA}$–$\mathcal{CID}$-based modules. Indeed, each module described in this section, except for NUM-RAC, was developed according to an open specification. That is, all possible sequences of navigation through the concepts and related information were implemented in the module, and which sequence to follow could be dynamically defined during its presentation. Such specific characteristic is responsible for guaranteeing flexibility to the modules. As a result, the same module can be adequate to different course formats without having to modify its structure significantly. For instance, although ITONCODE has been applied as an one-semester course, at an academic institution, it can also be used as a short-course, a tutorial or an invited talk, in scientific events; or as a training course, at industrial organizations. Exceptionally for NUM-RAC, the instructors asked for a close specification in order to keep the learners (in this case, children) better localized.

Finally, an interesting data to be analyzed refers to the effort spent (man-month) in the development/re-engineering process of the $\mathcal{IMA}$–$\mathcal{CID}$-based modules. In average, the effort values were low, varying from 2 to 0.25 man-month. The highest values were observed for the modules related to V&V domain, i.e., SOFTTEST, ITONCODE and JABUTI-TT. Actually, such modules address the subject domain in a more complete way in comparison with the other ones, dealing the a higher number of concepts, information items and instructional elements. SOFTTEST, particularly, corresponds to the first application of $\mathcal{IMA}$–$\mathcal{CID}$ and for this reason its development shows the highest effort. Since ITONCODE and JABUTI-TT reused parts of SOFTTEST, the effort spent on their development was lower. Besides that, it is important to observe that most of the effort related to JABUTI-TT development was spent on creating videos and animations for the testing tool and it is not associated with the development of the $\mathcal{IMA}$–$\mathcal{CID}$ models.

Regarding the other modules (SM-VTM, TECHINTER and OMM), the effort spent was low, even considering the non $\mathcal{IMA}$–$\mathcal{CID}$ specialists. Indeed, no significant difference in terms of effort could be noticed among domain experts and $\mathcal{IMA}$–$\mathcal{CID}$ specialists. Of course, if the author does not know $\mathcal{IMA}$–$\mathcal{CID}$, there is an effort to be spent for managing the modeling approach fundamentals. On the other hand, if the author is not an expert on the knowledge domain, there is also an effort for dealing with such related information. Particularly for the modules we have produced, the effort values were the same in both situations. We intend to better investigate this aspect in the next applications of $\mathcal{IMA}$–$\mathcal{CID}$.

Although preliminary, the results obtained so far are very positive in terms of the authors' attitudes toward applying $\mathcal{IMA}$–$\mathcal{CID}$. As further work, we intend to conduct systematic and controlled experiments aiming at providing more evidence in this direction.

6.2 SOFTTEST evaluation

To provide a preliminary evaluation on the SOFTTEST effectiveness, it was applied as part of a three-hour short-course on software testing for a group of about 60 undergraduate students with previous knowledge of software engineering [4]. We focused on theoretical aspects of testing, providing an introductory perspective on this subject. Practical aspects were illustrated but, due to time constraints, there was no direct participation by the audience. The effects of our approach were informally evaluated by applying a voluntary survey to the students after they had finished the course.

Considering a module developed without a systematic approach to structure the content (i.e., ad hoc development) as a traditional educational module, and a module developed by using $\mathcal{IMA}$–$\mathcal{CID}$ as a non-traditional educational module, the main research question we were interested in answering was: "*Learners' attitude toward accepting non-traditional educational module is more positive than toward accepting traditional one*?".

In this matter, the survey was composed of three main sections, covering the learners' attitude toward: (1) content, regarding to the concepts, additional information, examples and exercises used; (2) navigational aspects, focusing on the adoption of open specifications; and (3) general aspects about the module. Sections 1 and 2 were composed of objective questions while Sect. 3 consisted of subjective questions.

Regarding the content, the learners pointed out as positive aspects the way the module was structured and how it addressed the topics discussed. The connections between concepts were highlighted and the examples and additional information seemed to be appropriate. In terms of the proposed exercises, we noticed some expectation for practical tasks where the learners could actively participate. Although practical exercises involving the use of testing tools had already been integrated to the module, the short time available to the course made them trackless. The results pointed to the need for more concise exercises, which can be explored in such particular type of course.

Considering the navigational aspects, we observed a positive attitude toward the flexibility on choosing the sequences of presentation. Despite the large amount of information available, the students did not "get lost" in the module. Finally, regarding the specific characteristics of the module, aspects such as usability, instructor's energy, enthusiasm and objectiveness were also reported.

Besides the learners' evaluation, some instructor's responses were also observed by his comments after using the module. The possibility of having defined the sequences of navigation through the module during the "execution time", based on the learner's understanding and feedback, was an important point highlighted.

Although preliminary, one significant result observed by applying SOFTTEST was the very positive attitude, from the students and from the instructor, toward the flexibility provided by the module. In case of students, particularly, even without an active participation on using the module, they were able to realize the different possibilities of navigation explored by the instructor. Such flexibility, achieved by modeling the content as an open specification, was considered the key factor for better motivating and engaging students (and the instructor as well) in the course.

6.3 ITONCODE: evaluating the learning effectiveness

The previous-mentioned evidence motivated us to address a second research question, carried out in the context of experimental software engineering, more specifically in the scope of lab packages for evaluating V&V techniques (code inspection and testing): "*If subjects were given training using non-traditionally produced educational module would behave more uniformally, in the sense of fault detection rate, than if they were given training using traditionally produced module*?".

In this matter, we have replicated an extended version of the Basili & Selby experiment [7, 38], originally used for comparing V&V techniques, now considering the educational setting. Shortly, testing specialists and $\mathcal{IMA}$–$\mathcal{CID}$ experts have re-engineered the lab package training material on V&V techniques by applying the $\mathcal{IMA}$–$\mathcal{CID}$ approach [6]. The result was the ITONCODE module, which was briefly described in Sect 6.1.

The learning effectiveness was evaluated by the students' ability and uniformity on: (1) detecting existing faults; (2) generating test cases; and (3) covering the test requirements. Although in our experiment we have collected data for all these metrics, in this paper we focus on the benefits of the educational module produced by analyzing the student's uniformity in detecting existing faults, in the sense of measuring the percentage of faults each subject identified applying the involved techniques.

Aiming at answering our second question, the following hypotheses have been formulated:

- *H0*: There is no difference in the fault detection rates uniformity of subjects given training using non-traditionally produced module as compared to subjects given training using traditionally produced module.
- *Ha*: The fault detection rates uniformity of subjects given training using non-traditionally produced module are higher compared to subjects given training using traditionally produced module.

We applied ITONCODE in two one-semester undergraduate courses at ICMC/USP: *Exp*1 and *Exp*2 correspond to the experiments applied to SCE-702 (Software Testing and Inspection) and SCE-221 (Verification, Validation and Software Testing) courses, respectively. The main goal of both courses was to explore the fundamentals of V&V. The training was given in expositive classes, exploring the theoretical aspects of code inspection and testing activities and related supporting tools. At the end of each class, practical exercises were proposed. *Exp*1 and *Exp*2 involved 36 (9 teams) and 52 (13 teams) students, respectively. For the sake of space, in this paper we present only the data obtained from *Exp*1. More details can be found in [6].

For developing the experimental study we used a lab package created for replication in experiments involving V&V techniques [7, 38]. From this lab package we extracted information with respect to the selected programs, existing faults, forms for data collection, and procedures for conducting the study.

Regarding the code inspection activity, we used the code reading technique [47]. In case of the software testing activity, we used the following criteria: (1) from functional technique—equivalence classes partitioning [47]; (2) from structural technique—all-nodes [47], all-edges [47] and all-potential-uses [37]; and (3) from error-based technique—mutation analysis [15]. We also considered the idea of incremental testing—strategy in which the positive aspects of different techniques are combined in an evolutive testing process. So, in *Exp*1 were applied, in separate, code reading (inspection technique—$T1$) and mutation analysis (error-based technique—$T2$). The third technique refers to the incremental testing ($T3$) which, in this case, was the combination of functional and structural criteria, applied in this order: equivalence classes partitioning, all-nodes, all-edges and all-potential-uses.

Similarly to the experiment of Basili & Selby [7], the three V&V techniques were combined with three different programs, all of them having faults. *Cmdline* analyzes a command line for syntactic and partially for semantic correctness (268 LOC and 9 faults). *Nametbl* reads commands from a file and processes them in order to test functions which implement a symbol table for a certain computer language (270 LOC and 8 faults). *Ntree* reads commands from a file and processes them in order to test functions which implement a tree in which each node can have any number of child nodes (244 LOC and 8 faults).

Table 4 presents the definition of *Exp*1 relating teams, techniques and programs. It also illustrates the total number of faults each team have detected for each technique; these data will be analyzed further in this section. All teams had only been trained by using the non-traditionally produced module (ITONCODE). In the previous studies, the subjects had only been trained by using the traditional-produced-module. Each team applied each technique only once, always considering a different program. The idea was to ensure that all techniques would be applied to all programs.

Table 5 shows the average number of test cases generated by the teams in order to satisfy the test requirements of $T2$ and $T3$ (code reading application does not involve the generation of test cases). The test cases were manually generated

Table 4 Total number of valid detected faults for techniques $T1$, $T2$ and $T3$

Teams	*ntree* (8 faults)			*cmdline* (9 faults)			*nametbl* (8 faults)		
	$T1$	$T2$	$T3$	$T1$	$T2$	$T3$	$T1$	$T2$	$T3$
$G1$	–	1 (12.5)	–	–	–	5 (55.5)	3 (37.5)	–	–
$G2$	–	3 (37.5)	–	–	–	3 (33.3)	2 (25.0)	–	–
$G3$	–	5 (62.5)	–	–	–	9 (100.0)	7 (87.5)	–	–
$G4$	–	–	8 (100.0)	5 (55.5)	–	–	–	4 (50.0)	–
$G5$	–	–	0 (0.0)	3 (33.3)	–	–	–	5 (62.5)	–
$G6$	–	–	1 (12.5)	2 (22.2)	–	–	–	4 (50.0)	–
$G7$	1 (12.5)	–	–	–	4 (44.4)	–	–	–	5 (62.5)
$G8$	1 (12.5)	–	–	–	3 (33.3)	–	–	–	5 (62.5)
$G9$	5 (62.5)	–	–	–	3 (33.3)	–	–	–	3 (37.5)

Table 5 Average number of generated test cases by techniques $T2$ and $T3$

Experiment	*Technique*	*cmdline*	*ntree*	*nametbl*
*Exp*1	Mutation analysis—$T2$	217.7	23.0	18.3
	Incremental—$T3$	38.0	14.7	17.0

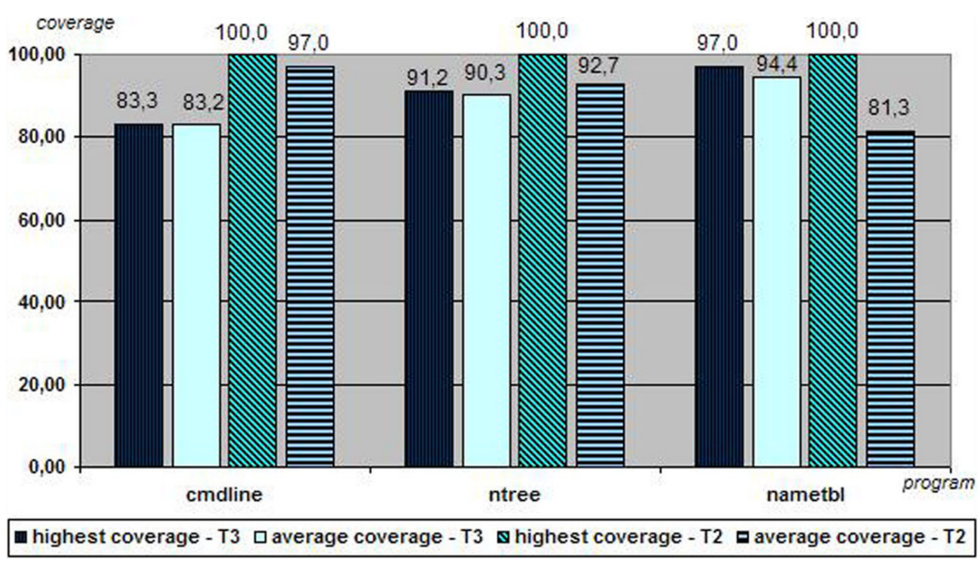


Fig. 13 Coverage analysis for techniques $T2$ and $T3$

based on the programs' specification and on the test requirements of each technique. For instance, consider the program *cmdline*. To satisfy $T2$ (mutation analysis), 217.7 test cases were generated, in average, by the three teams responsible for its application.

Figure 13 shows the coverage obtained by the teams, in average, with respect to the application of the generated test cases against $T2$ and $T3$. The highest coverages that could be obtained for each program and each technique are also presented. For instance, consider the program *cmdline*. The highest coverage for the techniques $T2$ and $T3$ would be 100.0% and 83.3%, respectively. The teams obtained, in average, a coverage of 97.0% for $T2$ and 83.2% for $T3$. Analyzing the table, we noticed that the teams reached adequate coverages and, in most of the cases, near to the highest possible values. Besides that, the teams also identified equivalent mutants for $T2$ and non-executable associations for $T3$. These activities had to be manually performed, being considered difficult, specially for "testing beginners".

Data on the total number of faults each team detected and which of them were valid or false-positive based on the existing faults have also been collected. Figure 14 shows the average number of valid detected faults by the teams with respect to the existing faults in each program, considering techniques $T1$, $T2$ and $T3$. For instance, for *cmdline*, from the 9 existing faults, the teams responsible for applying $T1$, $T2$ and $T3$ identified, in average, 3.33, 3.33 and 5.33 faults, respectively. Notice that these values refer to the applica-

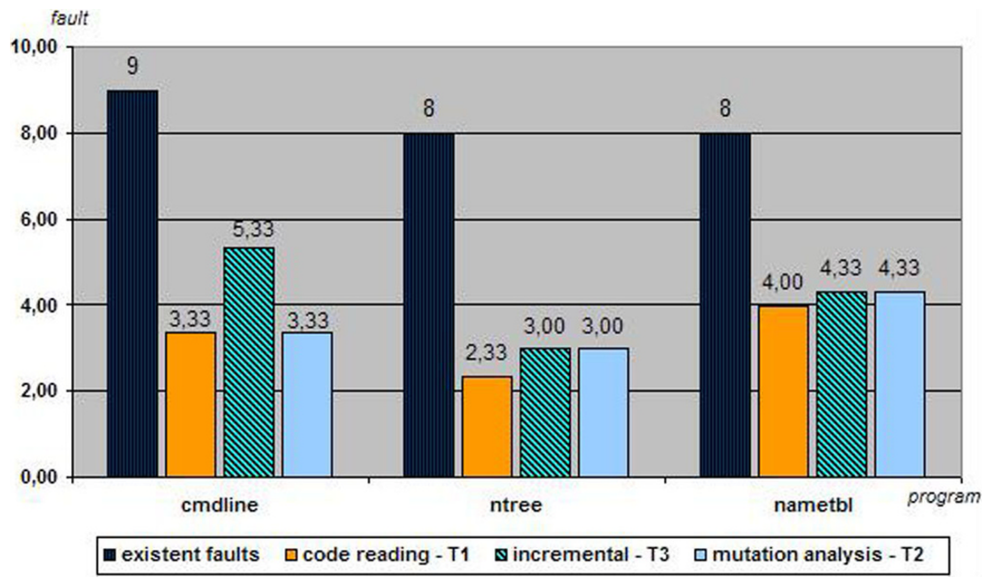


Fig. 14 Average number of valid detected faults

tion of each technique in separate. Analyzing the techniques together, we observed that all the 9 existing faults were identified by the teams. The same analysis was performed to the other programs; we noticed that most of the existing faults in the programs were detected when considering the application of all techniques together.

The data regarding the fault detection rates uniformity have been organized per team and per technique. Table 4 illustrates the total number of faults each team detected for techniques $T1$, $T2$ and $T3$. We have also analyzed the false-positives but they are not reported in this paper.

Any team with a fault detection rate over a pre-established threshold will be considered to have reached a satisfactory performance. We consider the behavior between two groups uniform if both overcome this pre-established threshold. In our study we consider a threshold of 30%; i.e., discovering over 30% of total faults.

In the previous experiments using traditionally produced modules it has been observed that for all the techniques, but for code reading, the behavior of the subjects was not uniform; e.g., only 50% of the subjects reached the threshold. From Table 4, we notice that for all the techniques, the behavior of the subjects was uniform: 55.5%, 88.8% and 77.7% of the subjects have reached the threshold for $T1$, $T2$ and $T3$, respectively. Actually, the student's uniformity in detecting faults seems to be better when using ITONCODE module.

The obtained results provided us preliminary evidence that hypothesis $H0$ can be refuted, meaning that the fault detection rates uniformity of subjects given training using non-traditionally produced module (in our case, ITONCODE) are higher compared to subjects given training using traditionally produced module.

As a final remark, we highlight that experiment *Exp*1 has some threats that might have an impact on the validity of the results. For instance, in the previous experiments the subjects were individuals and the time given to the training was very concentrate (around 6 hours). On the other hand, in *Exp*1 the subjects were considered as teams of individuals and the training was given in one-semester course (the students had more time to "mature" the V&V techniques).

Based on the results obtained so far we are motivated to conduct more systematic and controlled experiments to validate the ideas presented herein. In short-term, we intend to replicate *Exp*1 in order to compare the learning effectiveness in different scenarios: (1) training V&V techniques using traditionally produced module (ad hoc development); (2) training V&V techniques using non-traditionally produced module ($\mathcal{IMA}$–$\mathcal{CID}$-based module); and (3) training V&V techniques using SCORM-compliant module. In long-term, we intend to conduct experiments involving different courses and knowledge domains, offered to graduate and undergraduate students as well as to professionals from local industries. Content developers, learners and instructors' attitudes toward $\mathcal{IMA}$–$\mathcal{CID}$ and the produced modules should be evaluated.

7 Conclusions and further work

In this paper we provided a discussion of supporting mechanisms for the development of educational modules, focusing on the main characteristics of $\mathcal{IMA}$–$\mathcal{CID}$—an integrated approach for modeling learning content. Also, the application of $\mathcal{IMA}$–$\mathcal{CID}$ was illustrated in the development of educational modules for different knowledge domains, especially for software testing (SOFTTEST) and code inspection and testing techniques (ITONCODE).

The main contribution of our paper is to motivate the use of innovative mechanisms for creating well-designed, flexible and high-quality educational modules, which would provide: (1) transferability to different institutions and learning environments; and (2) effective support to traditional learning approaches, engaging learners in an empowering way; and (3) effective support to non-traditional environments, motivating the transition from lecture-based to active and lifelong learning.

Many issues regarding content modeling and the development of educational modules remain opened and must be further addressed. For instance, one of the perspectives we are now investigating refers to the use of ontologies [65] as supporting mechanisms for modeling the learning content. The goal is to evolve $\mathcal{IMA}$–$\mathcal{CID}$ to allow that both conceptual mapping and ontologies can be used for structuring and representing the knowledge domain. By using ontologies in the conceptual level of $\mathcal{IMA}$–$\mathcal{CID}$ we intend: (1) to provide a better comprehension of the knowledge domain to be taught; (2) to ease collaboration and knowledge sharing among authors; (3) to provide a well-established structure for a knowledge repository; and (4) to provide support for interoperability, considering the relationship among different paradigms and languages. Besides that, the adoption of ontologies in the didactical level should also be explored together with the idea of open specifications, aiming at providing knowledge reuse in different learning contexts. The input of learners in the very early stages of the module development, similarly to the participative approach in software development, should also be investigated.

Another matter for further investigation is related with automated learning environments and their support for content modeling. The fast evolution of information and communication technologies has significantly increased the number of learning environments available. In summary, such environments provide: (1) the required infrastructure for integrating the learning materials and for delivering/publishing them to the learners; (2) support to perform practical tasks and evaluations; and (3) support collaborative work and augment communication and discussion among instructors and learners. However, no mechanism for modeling the related knowledge domain is provided. Indeed, in most of the cases, the activity of content modeling is left in charge of the author, without any systematization. At most, some support for the storage and retrieval of learning content is provided.

In this sense, we have worked on the development of *IMA-Tool*—a supporting tool for the edition, interpretation and execution of the $\mathcal{IMA}$–$\mathcal{CID}$ models, providing mechanisms to simulate and validate executable specifications of the content. Indeed, applying $\mathcal{IMA}$–$\mathcal{CID}$ without an automated support can be an error-prone activity; additionally, the lack of automated tools for content modeling represents a constraint for its adoption.

IMA-Tool is an online collaborative tool for helping the "open" construction of the $\mathcal{IMA}$–$\mathcal{CID}$ models [9]. Based on the $\mathcal{IMA}$–$\mathcal{CID}$ models, *IMA-Tool* also provides support for content generation. In this automated scenario, we are now exploring the adoption of ontologies. As a first result, by using *IMA-Tool* the user can choose among ontologies and concept maps as the supporting mechanism for modeling the concepts. Besides that, we are also investigating the translation of $\mathcal{IMA}$–$\mathcal{CID}$ models into machine-readable specifications, automatically or by hand, in order to facilitate interoperability and promote reusability [61].

Based on the results obtained so far, we are motivated to conduct more systematic and controlled experiments involving the $\mathcal{IMA}$–$\mathcal{CID}$ application in order to validate the ideas presented in the paper. In short-term, we intend to conduct the $\mathcal{IMA}$–$\mathcal{CID}$ confrontation with other approaches for developing educational modules, mainly SCORM and IMS LD.

Also, we intend to investigate the development of educational modules to be applied in non-traditional environments (digital TV, tablets, mobile devices, and so on), focusing on the e-learning requirements and perspectives. Particularly, we are interested in explore how the idea of "open specifications" can be used for establishing a broad pedagogical model, capable of encompassing different strategies for active learning. Depending on the learner's cognitive style, an adequate strategy for presenting and navigating through the material would be selected.

At the very end, we intend to establish a culture for "open and collaborative learning materials" so that the use and evolution of them by a broader learning community would be better motivated and become a reality. The existence of a well-defined approach to systematize the development of learning content and, at the same time, flexible enough to be adaptable to different knowledge domains and development teams, plays a key role in crossing international, cultural and social borders in order to prepare the learners to be successful in a global market, with diverse groups of people.

Acknowledgements The authors would like to thank the Brazilian funding agencies (FAPESP, CAPES, CNPq) and to the *QualiPSo* Project (IST-FP6-IP-034763) for their support, as well as to the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

1. Advanced Distributed Learning Initiative (ADL): Sharable Content Object Reference Model (SCORM) specifications (2006). Retrieved April 10, 2011, from http://www.adlnet.org/
2. Ausubel DP, Novak JD, Hanesian H (1978) Educational psychology: a cognitive view, 2nd edn. Hold, Rinehart & Winston, New York
3. Barbosa EF (2004) Uma contribuição ao processo de desenvolvimento e modelagem de módulos educacionais. Ph.D. thesis, ICMC-USP, São Carlos, SP. In Portuguese

4. Barbosa EF, Maldonado JC (2006) An integrated content modeling approach for educational modules. In: IFIP 19th world computer congress—international conference on education for the 21st century, Santiago, Chile, pp 17–26
5. Barbosa EF, Maldonado JC (2011) Collaborative development of educational modules: a need for lifelong learning. In: Magoulas GD (ed) E-infrastructures and technologies for lifelong learning: next generation environments, pp 175–211. IGI Global, Birkbeck College, University of London, UK
6. Barbosa EF, Souza SRS, Maldonado JC (2008) An experience on applying learning mechanisms for teaching inspection and software testing. In: 21st conference on software engineering education and training (CSEET 2008), Charleston, SC, pp 189–196
7. Basili V, Selby RW (1987) Comparing the effectiveness of software testing strategies. IEEE Trans Softw Eng SE-13(12):1278–1296
8. Blackboard I (2002) Blackboard learning system (release 6): Product overview. White paper (http://www.blackboard.com)
9. Borges VA, Maldonado JC, Barbosa EF (2011) Towards the establishment of supporting mechanisms for modeling and generating educational content. In: 26th symposium on applied computing (ACM SAC 2011)—track intelligent/innovative learning environments, TaiChung, Taiwan, pp 1–6
10. Brachman RJ, Levesque HJ (eds) (1985) Readings in knowledge representation. Kaufmann, San Francisco
11. Brotherton JA, Abowd GD (2004) Lessons learned from eClass: Assessing automated capture and access in the classroom. ACM Trans Comput-Hum Interact 11(2):121–155
12. Caeiro-Rodríguez M (2007) Learning objects and instructional design: From contents to activities. In: Koohang A, Harman K (eds) Learning objects and instructional design, pp 219–251. Informing Science Press, Santa Rosa
13. (DCMI) (2005) D.C.M.I.: DCMI metadata terms. Retrieved 10 April, 2011, from http://dublincore.org/documents/2005/06/13/dcmi-terms/
14. Delamaro ME, Maldonado JC, Mathur AP (2001) Interface mutation: an approach for integration testing. IEEE Trans Softw Eng 27(3):228–247
15. DeMillo RA, Lipton RJ, Sayward FG (1978) Hints on test data selection: Help for the practicing programmer. IEEE Comput 11(4):34–43
16. Dewald NH (1999) Web-based library instruction: what is good pedagogy? Inf Technol Libr 18(1):26–31
17. Dick W, Carey L, Carey JO (2001) The systematic design of instruction, 5th edn. Addison-Wesley, New York
18. Dieberger A, Guzdial M (2003) CoWeb—experiences with collaborative web spaces. In: Lueg C, Fisher D (eds) From usenet to coWebs: interacting with social information spaces, 1st edn. Springer, Berlin
19. DotLRN (2009) DotLRN: open source learning management system. (http://dotlrn.org/)
20. Dougiamas M (2009) Moodle—a free, open source course management system for online learning. (http://moodle.org/)
21. EML: Educational modeling language (2000). Retrieved March 15, 2011, from http://hdl.handle.net/1820/81
22. Farrell K, Carr AE (2007) A blended model of instructional design for learning objects. In: Koohang A, Harman K (eds) Learning objects and instructional design, pp 359–405. Informing Science Press, Santa Rosa
23. Gagné RM, Briggs L, Wager W (1992) Principles of instructional design, 4th edn. HBJ College Publishers, Fort Worth
24. Gibbons AS, Bunderson CV, Olsen JB, Rogers J (1995) Work models: still beyond instructional objectives. Mach-Mediat Learn 5(3–4):221–236
25. Goldberg MW, Salari S, Swoboda P (1996) World wide web—course tool: an environment for building WWW-based courses. Comput Netw ISDN Syst 28(7–11):1219–1231
26. Harel D (1987) Statecharts: a visual formalism for complex systems. Sci Comput Program 8:231–274
27. Horn RE (1989) Mapping hypertext. Lexington Institute, Lexington
28. IEEE Learning Technology Standards Committee (LTSC) (2002) Learning Object Metadata (LOM)
29. IMS Global Learning Consortium Inc (2003) IMS learning design information model, final specification. Retrieved March 15, 2011, from http://www.imsglobal.org/learningdesign
30. IMS Global Learning Consortium Inc (2004) IMS content packaging, final specification. Retrieved March 15, 2011, from http://www.imsglobal.org/packaging
31. International organization for standardization (2004) Information technology—software life cycle processes. Standard. ISO/IEC 12207:1995/Amd 2:2004, ISO/IEC
32. Jonassen D (1999) Designing constructivist learning environments. In: Reigeluth CM (ed) Instructional theories and models, Mahwah, NJ, pp 215–239. Lawrence Erlbaum Associates
33. Kawasaki EI, Fernandes CT (1996) Modelo para projeto de cursos hipermídia. In: VII Simpósio Brasileiro de informática na educação, pp 227–240. Belo Horizonte, MG
34. Leinonen T, Purma J, Poldoja H, Toikkanen T (2010) Information architecture and design solutions scaffolding authoring of open educational resources. IEEE Trans Learning Technol 3(2):116–128
35. Leinonen T, Toikkanen T, Silfvast K (2008) Software as hypothesis: research-based design technology. In: 10th participatory design conference (PDC 2008), Bloomington, IN
36. Maidantchik CLL, Rocha AR (2002) Managing a worldwide software process. In: Workshop on global software development—international conference on software engineering (ICSE 2002), Orlando, FL
37. Maldonado JC (1991) Critérios potenciais usos: Uma contribuição ao teste estrutural de software. Ph.D. thesis, DCA/FEEC/UNICAMP, Campinas, SP
38. Maldonado JC, Fabbri SCPF, Mendonça M, Dória E, Martimiano LAF, Carver J (2006) Comparing code reading and testing criteria. In: ISESE 2006—international symposium on empirical software engineering, Rio de Janeiro, pp 42–44
39. Mayorga JI, Verdejo MF, Rodríguez-Artacho M, Calero MY (1999) Domain modelling to support educational web-based authoring. In: TET 99 congress, Norway
40. McConnell S (1999) Open-source methodology: ready for prime time? IEEE Softw 16(4):6–8
41. McGreal R (2010) Approaches to oer development. In: 7th open education conference. Retrieved April 10, 2011, from http://hdl.handle.net/10609/5083
42. McManus TF (1996) Delivering instruction on the world wide web. Online document. Retrieved March 15, 2011, from http://www.svsu.edu/mcmanus/papers/wbi.html
43. Memmel M, Ras E, Jantke K, Yacci M (2007) Approaches to learning object oriented instructional design. In: Koohang A, Harman K (eds) Learning objects: theory, praxis, issues, and trends, pp 281–325. Informing Science Press, Santa Rosa
44. Merrill MD (1983) Component display theory. In: Instructional design theories and models: an overview of their current states. Lawrence Erlbaum, Hillsdale
45. Merriënboer JV (1997) Training complex cognitive skills: a four-component instructional design model for technical training. Educational Technology Publications, Englewood Cliffs
46. Michener ER (1978) Understanding understanding mathematics. Cognitive Science 2(4)
47. Myers GJ, Sandler C, Badgett T, Thomas TM (2004) The art of software testing, 2nd edn. Wiley, New York
48. Novak JD (1990) Concept mapping: a useful tool for science education. J Res Sci Teach 27:937–949

49. Pansanato LTE, Nunes MGV (1999) Autoria de aplicações hipermídia para ensino. Revista Brasileira de informática na educação pp. 103–124
50. Paquette G, Teja I, Léonard M, Lundgren-Cayrol K, Marino O (2005) An instructional engineering method and tool for the design of units of learning. In: Koper R, Tattersall C (eds) Learning design: A Handbook on modeling and delivering networked education and training. Springer, The Netherlands
51. Pimentel MG (1997) MAPHE: Metodologia de apoio a projetos de hipertextos educacionais. In: VIII simpósio Brasileiro de informática na educação (SBIE 97), São José dos Campos, SP, pp 351–368
52. Pimentel MG, Ishiguro Y, Kerimbaev B, Abowd GD, Guzdial M (2001) Supporting long-term educational activities through dynamic web interfaces. J Interact Comput 13(3):353–374
53. Reigeluth CM (1999) The elaboration theory: audience for scope and sequence decisions. In: Reigeluth CM (ed) Instructional design theories and models: a new paradigm of instruction theory. Lawrence Erlbaum Associates, Hillsdale
54. Richards G, Marshall S (2010) Open educational resources for development of university courses. In: 7th open education conference. Retrieved April 10, 2011, from http://hdl.handle.net/10609/4846
55. Rodríguez-Artacho M (2002) PALO language overview. Tech rep, UNED University, Madrid, Spain. STEED project
56. Rodríguez-Artacho M, Maíllo MFV (2004) Modeling educational content: the cognitive approach of the PALO language. Educat Technol Soc 7(3):124–137
57. Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W (1991) Object-oriented modeling and design. Prentice Hall International, Englewood Cliffs
58. Russell SJ, Norvig P (2003) Artificial intelligence: a modern approach, 2nd edn. Prentice-Hall, Upper Saddle River
59. Sakai Project (2009) Collaboration and learning environment for education. http://sakaiproject.org/
60. Shepard T, Lamb M, Kelly D (2001) More testing should be taught. Commun ACM 44(6):103–108
61. Silva MAG, Barbosa EF, Maldonado JC (2011) Model-driven development of learning objects. In: 41st Frontiers in education conference (FIE 2011). Rapid City, SD. 6p, Accept for publication
62. Support centre for open resources in education (SCORE) (2011) SCORE project. Retrieved April 10, 2011, from http://www8.open.ac.uk/score/
63. Turine MAS, Oliveira MCF, Masiero PC (1997) Designing structured hypertext with HMBS. In: VIII International ACM hypertext conference (Hypertext 97), Southampton, UK, pp 241–256
64. UNESCO international institute for educational planning (IIEP) (2011) UNESCO OER Community. Retrieved April 10, 2011, from http://oerwiki.iiep.unesco.org/index.php/Main_Page
65. Uschold M, Grüninger M (1996) Ontologies: principles, methods and applications. Knowledge Engineering Review 11:93–136. doi:10.1017/S0269888900007797
66. Villani SA (2007) Learning objects and instructional design: theories, methods and tools useful in creating effective e-learning environments. In: Koohang A, Harman K (eds) Learning objects: theory, praxis, issues, and trends, pp 371–416. Informing Science Press, Santa Rosa
67. Vincenzi AMR, Wong WE, Delamaro ME, Maldonado JC (2003) JaBUTi: a coverage analysis tool for java programs. In: XVII Simpósio brasileiro de engenharia de software (SBES 2003), Manaus, AM
68. Wiley DA (2001) The instructional use of learning objects. Agency for instructional technology and association for educational communications and technology, Bloomington, IN. Retrieved March 20, 2011, from http://reusability.org/read/chapters/wiley.doc