

An Integrated Content Modeling Approach for Educational Modules

Ellen Francine Barbosa and José Carlos Maldonado
Computer Science Department, ICMC/USP – São Carlos
P.O. Box 668 13560-970 – São Carlos, SP, Brazil
{francine, jcmaldon}@icmc.usp.br

Abstract. Educational modules – concise units of study capable of integrating theoretical/practical contents and supporting tools – can be seen as relevant mechanisms to facilitate the student’s apprenticeship. The establishment of processes and modeling approaches should ease the cooperative work to create, reuse and evolve educational modules, taking also into account the impact on the learning process. There are initiatives to address the problem of modeling educational contents, but none of them provides a complete set of features addressing the conceptual, instructional and didactic perspectives. Moreover, these initiatives do not consider a systematic process for developing educational modules. In this work we summarize the main aspects of a standardized process for developing educational modules we have proposed, focusing on the modeling activity for structuring the learning contents. An integrated modeling approach is presented and its application is illustrated by the development of an educational module for the software testing domain. The material produced has been applied and preliminarily evaluated in terms of the student’s attitude toward content, usability and navigational aspects.

1 Introduction

Educational modules, which correspond to concise units of study delivered to learners by using technological and computational resources, can be seen as relevant mechanisms in order to ease the learning processes [1]. Besides that, the educational modules should be evolvable, reusable and adaptable to different learning scenarios and objectives. There are initiatives to address the problem of modeling educational module contents¹. In fact, these initiatives aim at providing ways to establish effective educational products capable of creating and/or improving motivational learning situations, but none of the initiatives provides a complete set of features addressing the conceptual, instructional and didactic perspectives. Moreover, they do not consider a process for developing educational modules.

¹ An overview of modeling approaches for educational contents can be found in [1].

Please use the following format when citing this chapter:

Barbosa, E.F., Maldonado, J.C., 2006, in *International Federation for Information Processing*, Volume 210, *Education for the 21st Century-Impact of ICT and Digital Resources*, eds. D. Kumar, and Turner J., (Boston: Springer), pp. 17–26.

We investigate the establishment of a process for developing educational modules, aiming at providing a set of guidelines and supporting mechanisms to create, reuse and evolve them [1]. Particularly, we are interested in the content-modeling activity, which helps the author to determine the relevant parts of the knowledge domain, providing a systematic way to structure the concepts, also promoting reusability and adaptability. In our research line, at the very end, we intend to provide a context for “open learning materials”.

In this paper we focus on the content-modeling activity, presenting the main aspects of an integrated modeling approach for developing educational content (*IMA-CID: Integrated Modeling Approach – Conceptual, Instructional and Didactic*). We illustrate the practical application of *IMA-CID* by the development of an educational module for the software testing knowledge domain. The testing material produced has been applied and preliminarily evaluated in terms of the student’s attitude toward content, usability and navigational aspects.

The remainder of this paper is organized as follows. In Section 2, we provide a brief overview of educational modules. In Section 3 we summarize the main aspects of a standardized process for developing educational modules we have previously proposed and describe the main aspects of *IMA-CID*. Also, *IMA-CID* is applied to the development of a software testing educational module, which is described in Section 4. Some results from a very preliminary evaluation on the effectiveness of *IMA-CID* are discussed in Section 5. Finally, in Section 6 we summarize our contributions and discuss the perspectives for further work.

2 Educational Modules

Educational modules are concise units of study, composed of theoretical and practical contents which can be delivered to learners by using technological and computational resources [1]. For theoretical contents we understand books, papers, web information, slides, class annotations, audio, video, and so on. Practical contents are the instructional activities and evaluations, and their resulting artifacts (e.g., executable programs, experiments, collaborative discussions). Specific tools related to the knowledge domain as well as the results obtained from their application can also be seen as practical contents. In the case of testing, *Proteum* [4] is an automated tool that can be integrated in an educational module to enable the application of testing concepts in real situations, fostering education and training situations and promoting exchange of technology between industry and academia.

Theoretical and practical contents are integrated in terms of learning materials. Learning environments, presentation tools, and mechanisms to capture classroom lectures and to support discussion spaces and collaborative work are examples of the required infrastructure for delivering the learning materials.

The development of educational modules should consider the intrinsic characteristics of knowledge, such as its dynamic and evolutionary aspect. In the testing domain, for instance, practical activities involving the conduction of experimental studies can result in new knowledge on testing techniques and criteria, which should be incorporated to the contents previously defined. Also, there is a need for adaptability and reusability – educational modules should be seen as

independent units of study, subject to be adaptable and reusable in different education and training scenarios, according to some aspects such as the learner's profile, instructor's preferences, learning goals and course length, among others.

3 Developing Educational Modules

- **A Process for Developing Educational Modules**

Similar to software modules, educational modules require the establishment of systematic processes in order to produce quality products. The Standard Process for Educational Modules [1] is based on the ISO/IEC 12207, tailored to the context of educational modules by including aspects of content modeling, practices from instructional design, and issues of distributed and cooperative work. The standard establishes a set of processes that can be used to acquire, supply, develop, deliver, operate, and maintain educational modules. Primary processes deal with the main activities and tasks performed during the life cycle of an educational module. Supporting processes contribute to the success and quality of the development project. Organizational processes are employed by an organization to establish, implement and improve an underlying structure made up of associated life cycle processes and personnel. Aspects of specialization and instantiation have also been explored in order to apply the standard process into specific projects, for different knowledge domains. In the same line as CMMI for software, a maturity model for educational modules – *CMMI/Educational* – was proposed as a mechanism to support the specialization of the standard process in different maturity levels.

- **The IMA-CID Approach**

Content modeling plays a fundamental role in the development process of educational modules [1]. It helps the author to determine the main concepts to be taught, providing a systematic way to structure the relevant parts of the knowledge domain. How the contents are structured can also affect the reusability, evolvability and adaptability of the module. Despite its relevance, there are few approaches for modeling educational contents. Besides that, each of these initiatives deals with different perspectives, which can be suitable for a given learning scenario but inadequate for others.

Motivated by this scenario, in a previous work we proposed a preliminary set of modeling requirements [2]: (1) concepts taxonomy, (2) concepts composition, (3) specific relationships, (4) hierarchical decomposition, (5) knowledge categories, (6) learning contexts, (7) pedagogical order, (8) history mechanisms, and (9) event propagation. We also identified some modeling perspectives – conceptual, instructional and didactic – in order to characterize the models for representing educational contents [2, 1]. Based on the requirements and perspectives, we carried out comparisons among the existent modeling approaches. In short, we noticed that some approaches seem to be particularly interesting in dealing with conceptual issues, while others deal with relevant elements under the instructional perspective, and still others demonstrate an expressive power for representing didactic aspects.

Finally, we established a connection between the perspectives and requirements, which was the starting point for the definition of the models for educational contents

and their integration into *IMA-CID* (*Integrated Modeling Approach – Conceptual, Instructional and Didactic*) [2, 1] – an integrated approach for modeling contents, composed by a set of models, each one dealing with specific aspects of the development of educational contents (Figure 1).

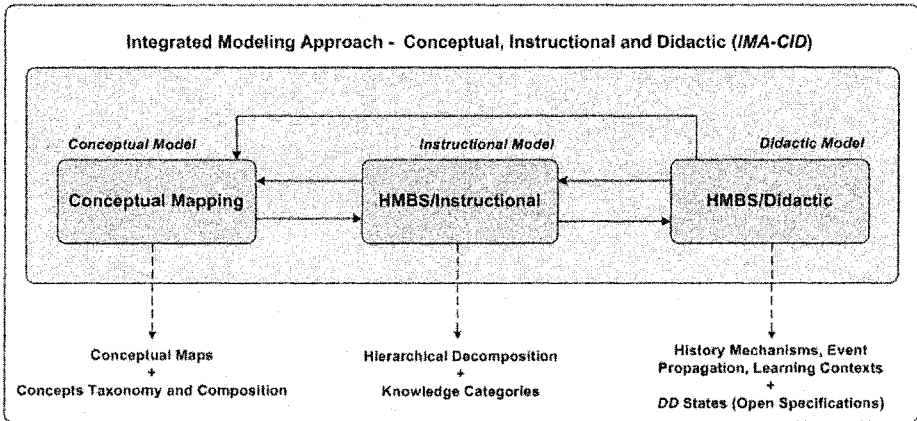


Figure 1. The IMA-CID approach

The *Conceptual Model* consists of a high-level description of the domain, representing its main concepts and the relationships among them. The relationships can be divided into two classes. Structural relationships (such as *type-of* and *part-of*) are useful to set up taxonomies among concepts and make inferences about the knowledge, representing a generic category of relationships, applicable to any kind of domain. Domain-specific relationships are user-defined and have their meaning associated with a particular subject, carrying their own semantics. They represent specific relations, whose interpretation depends on the domain being modeled.

To construct the conceptual model, we focused on the Conceptual Mapping ideas [9]. 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 existent modeling approaches for educational contents. We also included some additional notations to the rules for creating conceptual maps aiming at representing the relationships of concepts taxonomy (*type-of*) and concepts composition (*part-of*).

Besides concepts, information items and instructional elements should be considered as part of the knowledge domain. In the *Instructional Model* we are interested in defining such additional information, relating it to the concepts previously identified. Notice that we are not interested in how the information will be sequenced, but in what kind of information we can use to develop a more significant and motivating content. Several theories and techniques can be referenced to support the establishment of information items. We adopted the Component Display Theory [7], which specifies *concepts*, *facts*, *procedures* and *principles* as information items. The instructional elements can be classified into three categories. *Explanatory elements* (examples, hints, suggestions of study) deal

with the complementary information for explaining a topic. *Exploratory elements* (guided exercises, simulations, hands-on assignments) allow the learner to navigate through the domain, practicing the concepts and relevant information. *Evaluative elements* (diagnostic, formative and summative evaluations, in terms of subjective and/or objective questions) allow assessing the learner's proficiency on the domain.

As a support to construct the instructional model, we adopted the HMBS (Hypertext Model Based on Statecharts) model [10]. In short, HMBS uses the structure and execution semantics of Statecharts to specify the structural organization and the browsing semantics of hyperdocuments. We focused on the mechanisms for hierarchical decomposition HMBS provides, complementing the idea of hierarchical organization already explored in the conceptual model. Notice the hierarchy deals with the depth of the knowledge/material to be presented. To make HMBS suitable for modeling the instructional aspects, it was extended for representing different *knowledge categories* – concepts, information items and instructional elements. The extended version of HMBS is named *HMBS/Instructional*.

The *Didactic Model* is responsible for the establishment of prerequisites and sequences of presentation among conceptual and instructional elements. The specification of behavioral aspects can also be explored. Thus, the model 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. Didactic models are also 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 deals with relevant aspects under the didactic perspective (history mechanisms, event propagation and learning contexts definition), it was also adopted to construct the didactic model. In addition, by using HMBS we can validate the educational content through the analysis of the subjacent statechart properties [10]. As an extension to HMBS at the didactic level, we introduced the idea of *open specifications*, which provide 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. Moreover, when an educational module is implemented based on an open specification (*open implementation*), its navigation paths can be defined by the user himself, in "execution time". That is, during the presentation the user is able to dynamically decide which topics should be navigated and in which sequence, based on the learner's feedback for instance.

To represent open specifications, we extended HMBS with the notion of *DD* (Dynamically Defined) *states*. In short, all OR substates of a *DD* state (OR_{DD}) are totally connected to each other. That is, from any substate of a *DD* state X , we can reach all other substates of X . For the sake of legibility, transitions and events are implicitly represented. We also established a hierarchy of *DD*-superstates – leaving a *DD* state X can activate the OR_{DD} states from the hierarchy of *DD*-superstates of X . Both the notions of *DD* states as well as the hierarchy of *DD*-superstates help us to the establishment of open specifications in the sense that they allow us to represent

all sequences of presentation in the same didactic model. The extended version of HMBS to support *DD* states (and open specifications) is named *HMBS/Didactic*.

4 An Educational Module for Software Testing

IMA-CID was applied as part of the development process of an educational module for software testing (Figure 2). Concepts, facts, principles, procedures, examples and exercises were modeled and implemented as a set of slides, integrated into HTML pages, text documents, learning environments and testing tools. The module was designed and implemented according to the Standard Process for Educational Modules and the *IMA-CID* models [2, 1]. For the sake of space, in this section we present a specific model – the *HMBS/Didactic* –, developed for a particular subject of testing – the mutation analysis criterion [5]. The model is illustrated in Figure 3 and corresponds to an open specification.

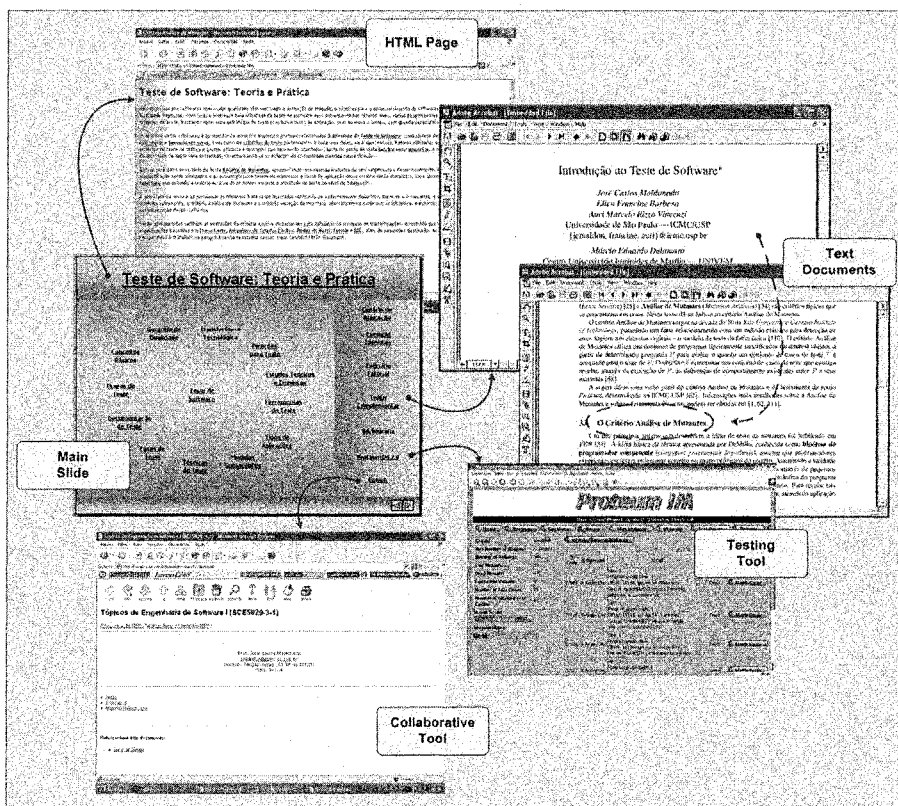


Figure 2. Educational module for software testing

Consider, for instance, the *MutationAnalysis* state. Besides the concept itself (*MA:concept:text*), some related facts (*MA:fact:text*) and principles (*CompetentProgrammer:principle:text* and *CouplingEffect:principle:text*) are specified too.

Consider now the `MutationAnalysisDetails` state. By exploring the notion of *DD* states, the `MutationAnalysisDetails` substates (*OR_{DD}* states) – `MutantOperator`, `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).

We can also explore the idea of a hierarchy of *DD*-superstates. Consider the sequence (`MutantGeneral`, `MutationAnalysisDetails`, `MutationAnalysisGeneral`, `ErrorBasedTechnique`, `TestingTechnique`, `SoftwareTesting` `TheoryPractice`) 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`, this analysis should be carried out for all states of the hierarchy. Notice we cannot get to the states `AlternativeApproaches` and `ApproachesClassification` from `Mutant`. Indeed, `ApproachesGeneral` does not pertain to the hierarchy of *DD*-superstates of `Mutant`.

Explanatory and exploratory elements were also represented. For instance, the concept of a mutant (`Mut:concept:text`) is illustrated by an example (`Mutant:example:figure`), which corresponds to an explanatory element. The exercise represented by the `ApplicationMA:exercise:text` state corresponds to an exploratory element, where mutation analysis is applied to test the factorial program. The required tools for doing the exercise are modeled too. The `Coweb:tool` state represents a collaborative environment (*CoWeb* [6]), used as a discussion space among learners and instructors. The `ProteumIM:tool` corresponds to a testing tool (*Proteum* [4]), used for applying the mutation analysis.

Besides the open specification, a *partially open specification* and a *closed specification* were also considered in order to define the didactic model for mutation analysis [1]. 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. Instead of having just implicit transitions, the idea is to make some of them be explicitly represented in the didactic model. In a closed specification, all sequences are predefined, that is, just a fixed sequence of presentation is available in the module. In this case, the transitions are explicitly represented. Notice the sequences of presentation derived from partially open specifications and from closed 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 Figure 3, several implementations of the same content about mutation analysis can be obtained. This characteristic is essential to generate differentiated contents (and modules as well), whose topics, depth and sequences of presentation are established according to some particular aspects (e.g., course length, pedagogical goals, instructor’s preference, learner’s profile).

The decision on which kind of specification to use should be based on the users (learners and instructors) and on the expected characteristics of the module. One

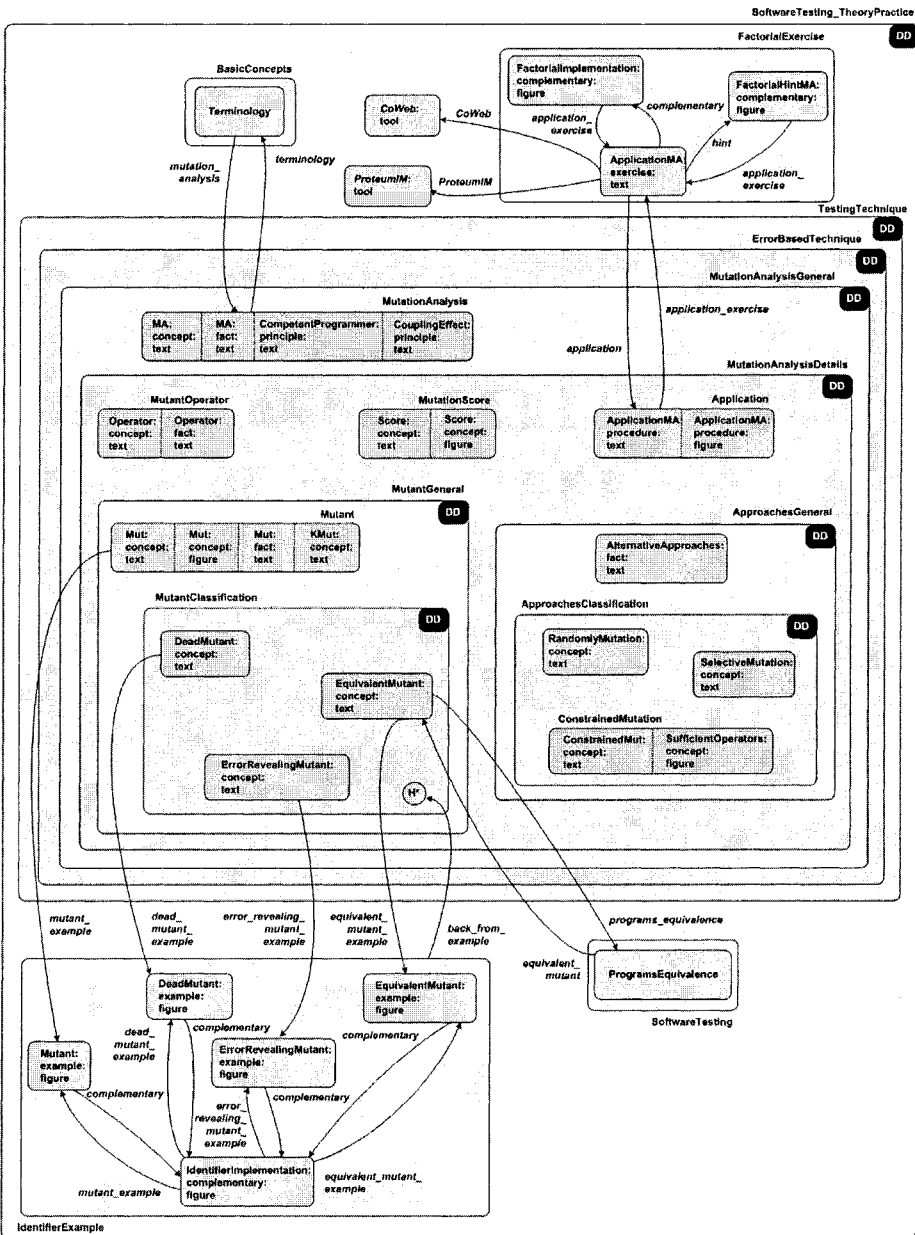


Figure 3. Didactic model (HMBS/Didactic) for mutation analysis

strength of open specifications is the flexibility to navigate 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. Indeed, while for less experienced instructors a closed specification (and implementation) seems to be the better choice, for the most experienced ones, an open specification would be an adequate alternative too.

5 Evaluation of the Educational Module for Software Testing

To provide a preliminary evaluation on the effectiveness of the testing module, we applied it in a three-hour short course on software testing for a group of 60 undergraduate students with some previous knowledge on software engineering. 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. The survey was composed of four sections, covering the student's attitude toward: (1) content, regarding the concepts, additional information, examples and exercises used; (2) usability, in terms of the interface; (3) navigational aspects; and (4) general aspects about the module. Sections 1, 2 and 3 were composed of objective questions while section 4 consisted of subjective questions.

Regarding the content, the students 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 appropriate. Regarding the proposed exercises, we noticed some expectation for practical tasks where the students could actively participate. Although practical exercises involving the use of testing tools had already been integrated into the module, the short time available in the course made them intractable in the context of the course. The results pointed to the need for more concise exercises that can be explored in this kind of course. In terms of usability, the schema of colors, the distribution of information through the pages/slides and the representation of the interface functions were, in general, well accepted by the students. Specific comments indicated some disappointment with respect to the size of fonts and figures. Regarding the navigational aspects, we observed a positive attitude toward the flexibility on choosing the sequence of presentation. Despite the large amount of information available, the students did not "get lost" in the module. Finally, aspects such as the instructor's energy, enthusiasm and objectiveness were also reported.

The results obtained provide some very preliminary evidence on the practical use of *IMA-CID* and its modeling mechanisms as a support to the development of effective educational modules. However, we highlight the need for conducting a more systematic and controlled experiment to validate our ideas. This experiment has been planned for the next term, involving three different courses on testing, offered to graduate and undergraduate students at ICMC/USP. Both students and instructors' attitudes toward the module should be evaluated.

6 Conclusions and Further Work

In this paper some mechanisms for developing educational modules were discussed and an integrated modeling approach (*IMA-CID*) was presented. Also, some points of the application of *IMA-CID* were illustrated by the development of an educational module for the testing domain. As further work, we intend to investigate the definition of supporting tools for the *IMA-CID* models. We are interested in automated tools for helping the interpretation and execution of the *HMBS/Didactic*

model, providing mechanisms to simulate and validate executable specifications of the content. Tools for automatic content generation should also be explored.

Moreover, we are motivated to keep investigating the mechanisms we have proposed in future offerings of testing courses. In this sense, we are now working on the development of an educational module for the integrated teaching of testing and programming foundations in introductory CS courses [3]. Since our mechanisms can be applied to different kinds of domains, we are also interested in using them to develop and evaluate educational modules for other areas.

Another perspective is to explore the development of learning objects under the context of educational modules. The idea is to apply our modeling mechanisms to structure, store and retrieve the internal components of these objects. Further studies have also been planned in order to investigate the use of conceptual models on the development of domain ontologies and vice versa. Finally, at the very end we intend to establish a culture for “open learning materials”, so that the use and evolution of them would be better motivated.

Acknowledgements

The authors would like to thank the Brazilian funding agencies – FAPESP, CAPES and CNPq – for their support to this research.

References

1. E. F. Barbosa. Uma Contribuição ao Processo de Desenvolvimento e Modelagem de Módulos Educacionais. *PhD thesis*, ICMC-USP, São Carlos, SP, Mar. 2004.
2. E. F. Barbosa, J. C. Maldonado, and I. L. M. Ricarte. Learning materials: Towards the establishment of guidelines for domain modeling. In *Informatics Curricula, TEaching Methods and best practice Working Conference*, Florianópolis, SC, July 2002.
3. C. Corte, E. Barbosa, and J. Maldonado. Integrated teaching of programming foundations and software testing. In *IFIP International Conference on Education for the 21st Century*, Santiago, Chile, 2006. (Poster session.)
4. M. E. Delamaro, J. C. Maldonado, and A. P. Mathur. Interface mutation: An approach for integration testing. *IEEE Trans. on Software Engineering*, 27(3):228–247, Mar. 2001.
5. R. A. DeMillo, R. J. Lipton, and F. G. Sayward. Hints on test data selection: Help for the practicing programmer. *IEEE Computer*, 11(4):34–43, Apr. 1978.
6. M. Guzdial, J. Rick, and C. Kehoe. Beyond adoption to invention: Teacher-created collaborative activities in higher education. *Journal of the Learning Sciences*, 2002.
7. M. D. Merrill. Component display theory. In *Instructional Design Theories and Models: An Overview of their Current States*, Hillsdale, NJ, 1983. Lawrence Erlbaum.
8. G. J. Myers, C. Sandler, T. Badgett, and T. M. Thomas. *The Art of Software Testing*. John Wiley & Sons, 2nd. edition, 2004.
9. J. D. Novak. Concept mapping: A useful tool for science education. *Journal of Research in Science Teaching*, 27:937–949, 1990.
10. M. A. S. Turine, M. C. F. Oliveira, and P. C. Masiero. Designing structured hypertext with HMBS. In *VIII International ACM Hypertext Conference*, pages 241–256, Southampton, UK, Apr. 1997.