

A Design Model for Educational Multimedia Software

André Koscianski, Denise do Carmo Farago Zanotto

UTFPR, Ponta Grossa, Brazil

Email: koscianski@utfpr.edu.br

Received 15 September 2014; revised 12 October 2014; accepted 10 November 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The design and implementation of educational software call into play two well established domains: software engineering and education. Both fields attain concrete results and are capable of making predictions in the respective spheres of action. However, at the intersection, the reports about the development of computer games and other educational software reiterate similar difficulties and an embarrassing degree of empiricism. This study aims to bring a contribution, presenting a model for the conception of educational multimedia software by multidisciplinary teams. It joins three elements: Ausubel's theory of meaningful learning; the theory of multimedia from Mayer; and the study of software ergonomics. The structure proposed here emerged from a theoretical study with the concurrent development of software. Observations gathered in a small scale test confirmed the expected design issues and the support provided by the model. Limitations and possible directions of study are discussed.

Keywords

Multimedia, Educational Software Design, Pedagogic Issues

1. Introduction

The first years of school are characterized by a ludic atmosphere that welcomes students into a warm and receptive ambience, with a careful mixture of leisure activities and serious objectives. Toys and games are practically mandatory elements from kindergarten up to K-12 classes. Some of the goals of such environments are to develop a positive attitude towards the school and cultivate reading and studying habits. Gradually there is a shift of perspective, culminating in the university, where courses are labour-intense and adopt a rather strict—if not spartan—style. Nonetheless, teachers of all educational levels tend to seek strategies to lower stress and keep students engaged and motivated.

Multimedia software can be helpful in this context for a number of reasons. These applications have native capabilities to show and interrelate textual descriptions, diagrams and photos, besides videos and sounds. All these means once combined represent a large contact surface between the previous knowledge of the student and the new information, helping to favour assimilation. The availability of images, animations and sound makes it possible to explore the presentation of material under ludic perspectives, counterbalance boredom and maintain the level of attention. In other terms, the students are presented to the same core information, although it is deliberately disguised only to change their internal disposition towards the subject. Finally, software can be designed to enhance the interaction between students and the learning materials, by means of activities that demand several inputs of different types.

The possibilities of using computers in the classroom are widely discussed in the literature, but the idiosyncrasies of educational software development are not addressed to the same extent. Using off-the-shelf products is just one facet of the introduction of technology in the classroom. Implementing such artefacts involves the design of interface and interaction which depend on a series of factors and unfold in a complex set of requirements. The project of such software should ideally be coupled to the instructional design of a discipline and not be limited to the design of a module. Aesthetic and artistic criteria also have a significant impact on the way students perceive the computer and get involved with it. Finally, the trade-off between the development cost and the utility of the product, as part of a set of comprising books, laboratories and other elements, is seldom discussed.

Several domains are interwoven in the design and construction of educational software and multimedia; three tools are considered here in order to structure the requirements and the design of such applications. The area of human-computer interaction and more particularly, the study of ergonomic interfaces helps to shape the overall design of applications and avoid mistakes that could negatively affect most users of a given audience. The Cognitive Theory of Multimedia Learning (CTML) of Richard Mayer is a landmark in the field of educational software, providing important heuristics aiming to optimize the effects of multimedia as instructional means. Finally, the Theory of Meaningful Learning (TML) of David Ausubel proposes explanations for the mechanisms behind learning and gives clues about how learning materials should be organized, so that software is integrated into a thorough instructional design. Taken together, this information can feed software engineering processes and provide references for the project and implementation of multimedia instructional programs.

This article discusses the design of multimedia software integrating these different views. An application was developed for the subject of biology, for Brazilian students with little contact with computers. Evaluations of a group of specialist and of the students are presented and discussed.

2. Organizational Issues in Educational Software Development

The project and implementation of any computer program is based on software engineering principles, which affect product quality and process efficiency (Pressman, 2009). The field covers a broad range of issues, varying from technical content like algorithmic complexity and hardware performance, to team management and psychological aspects involved in quality control (Weinberg, 1999). The development of educational software can be particularly complicated due to multitude of aspects to be considered, comprising cognitive and psychological effects as well as several technical problems like programming fast graphics or simulating physics to increase the realism.

Despite its relative youth, software engineering can be considered a mature field, being capable of coping with the construction and maintenance of all the complex informational infra-structure that surround us nowadays. To this end it has methodological tools to integrate specific expertises into the development of computer programs. The examples are abundant: medical tools, engineering programs, economics, chemistry and, of course, educational applications. However, working with multi-disciplinary teams may still be a daunting task. For instance, a typical programmer or software analyst is not used to the subtleties involved in the transmission of information during a teaching-learning interaction:

“Instead of guessing, designers should have access to a pool of representative users after the start of the design phase. Furthermore, users often raise questions that the development team has not even dreamed of asking” (Nielsen, 1992b).

Even if this statement is twenty years old, it remains a fresh truth in software engineering. Teachers are not acquainted with the technical barriers, nor the possibilities, that define the choice of a given set of functions or features that should be implemented in new software. Despite being well-documented in the informatics litera-

ture (Yourdon, 1989; Weinberg, 1999; Pressman, 2009), these issues are recurrent.

A minimum-team for educational software development would include a teacher, a programmer, a graphical artist and a manager, as these four persons cover the basic tasks and responsibilities associated with a project of this nature. The integration of experts in a team and the exchange of information must be constantly and actively supervised. Even small teams give rise to complex social interactions (Kirkman & Rosen, 1999). The project manager must balance different views (Weinberg, 1999).

One of the most critical aspects of software development is the initial phase of requirements elicitation (Pressman, 2009; Wiegers, 2003). Software behaviour, user expectations, hardware specifications are some of the characteristics that must be tailored to the application domain using information gathered from several sources, which include regular users, experts, technicians, investors and also standards and regulations. The boundaries separating the areas and the professionals are not very sharp (Weinberg, 1999; Flynt & Salem, 2005).

Concepts, ideas and knowledge from the domain of pedagogy enter as inputs to engineering processes. Inversely, constraints related to the computer and the resources available, may steer the project in a manner not foreseen by the teachers.

Final users and experts like teachers, psychologists and students can actively participate of the implementation, providing drawings, modifying the project of the interface, validating prototypes. **Figure 1** identifies this domain of action by calling it the “teacher space”, distinguishing it from the work of programmers and analysts, called here the “software engineering space”. There is a mutual influence between the areas that must be balanced throughout the implementation. For instance, a choice like presenting or not videos in the interface is directly related to instructional design expectations and needs. However, it will be up to the technicians to determine the feasibility of this requirement and, if possible, to project and create a solution. Eventually a request must be turned down on the basis of technical or resource limitations, this way making it necessary to reconsider the project in the teacher space. The inverse may also happen, opening possibilities that were not considered at the beginning. The gray gradient of the figure represents the interwoven nature of this relation between experts.

There are different ways to organize the implementation of a software project. They range from the management of small applications carried out with PSP (Personal Software Process) to projects counting hundreds of developers relying on methods as the CMMI (Capability Maturity Model Integration). Giving the intrinsic complexity of multimedia, the use of a methodology for development should not be underestimated, even in modest projects. Studying the available choices is out of the scope of this article; the interested reader could refer to Pressman (2009).

Multimedia applications share many characteristics with videogames. Both tend to contain numerous artefacts that include drawings, diagrams, sounds, music, storylines, descriptions of characters, texts and dialogues. The array of items may require specific management tools as versioning control and indexing (Flynt & Salem, 2005).

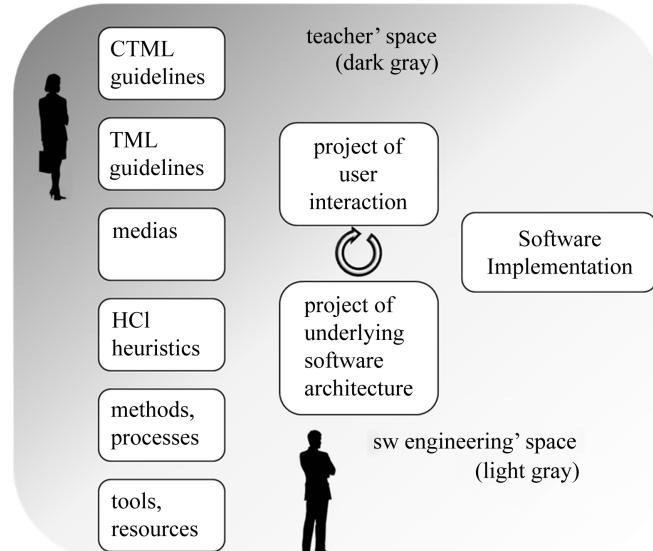


Figure 1. Educational multimedia design space.

A crucial, yet simple tool, often used in management by the videogame industry, is the Game Design Document. In its original form and purpose, it is a document making a description of the software from the user viewpoint, without much concern for the internal architecture (Bethke, 2003; Rollings & Morris, 2004). It describes artefacts as the layout of screens, sequencing of tasks and possible paths of interaction that have a clear interest in an educational setting. The document functions as a focal point for the development team, helping tasks as evaluation of possible designs, negotiation of priorities and definition of deadlines. Non-ambiguous, complete and accurate design documents avoid difficulties and problems that can permeate the project up to the final product implementation (Flynt & Salem, 2005; Pressman, 2009).

Complementary to the Game Design Document, storyboards and prototypes (Meigs, 2003; Johnson & Scheleyer, 2003) are invaluable for educational multimedia. They allow discussing the project with an eye on the finished product.

3. Interface Design and Educational Software

Human Computer Interaction (HCI) can be thought of as a subset of interaction design (Preece & Rogers, 2002). Interaction design is a field of research concerned with the interaction between humans and things like cell phones, control panels used in industries, airplanes cockpits or everyday objects.

The typical computer interface devices—keyboards, mice and touch screens—are, in a certain sense, very limited artefacts: if we observe users interacting with software, we see silent people staring at screens, making movements with their hands while their bodies rest immobile for long periods. All the richness of the tasks being executed is concealed in the cognitive interaction between the users and the elements that are presented on screen. In fact, the project of computer interfaces involves a high level of subjectivity (Nielsen, 1992a; Preece et al., 2002). As a consequence there is a rather paradoxical problem in software engineering: while the design and implementation processes are well known and reasonably controlled, the exact quality and behaviour of the final software is less certain, being highly dependent on a given context of use (Weinberg, 1999; Koscienski & Soares, 2006). Interfaces are, in this sense, blurry targets and the difficulty to forecast the success of a videogames is a good example of this (Bethke, 2003; Rollings & Morris, 2004).

The discipline of HCI studies the relationship between humans and computer interfaces from several viewpoints. For instance, the analysis of tasks offers a perspective to organize sequences of user actions, like pressing buttons and filling fields with information, in order to increase efficiency and prevent mistakes; the field of psychology gives clues on how to draw attention, avoid distractions and keep users focused; and research in physiology and cognition helps to understand how people react to stimuli and how they cope with tasks using information that is laid out on the screen. A straightforward example is to limit the branching factor and the number of items that are shown in menus. The famous work of Miller (1956) discussing short term memory is one of the firsts among a long series of research around cognitive capabilities.

Making a coherent unit out of all these pieces of information is a non-trivial task. The field of HCI has established criteria that helps guide software design and avoids most evident flaws (Nielsen, 1992a).

The international standard ISO 9241 is an important reference for interface design. It gives a series of advices concerning ergonomics, but leaves out the needs from specific domains like education. This issue has been addressed by several studies, some examples being Squires and Preece (1999); Ardito, Buono, Costabile, Lanziotti, & Piccinni (2009); Alsumait & Al-Osaimi (2009). The general character of the standard and its orthogonal organization make it possible to map virtually any set of requirements. **Table 1** illustrates this point with one of the studies specific to educational software.

The first two columns of **Table 1** lists the principles from ISO 9241, part 10; and the heuristics established by Nielsen (1992a) for usability of software. Both sets represent very condensed information that unfolds in more specific characteristics. For instance, in Safdari, Dargahi, Shahmoradi, & Nejad (2012) a questionnaire of 75 items was derived out of the seven ISO principles presented in the first column of the table. In the same manner, the requirement “conform to user expectations” is reworked by Alsumait and Al-Osaimi in several remarks, adapted to the universe of children education.

Besides covering all foundation of user interaction—a rich subject in itself—HCI should also accommodate instructional demands that are not part of most computer programs. Usual applications are designed to support tasks executed in a straightforward manner; examples are editing a letter, filtering information in databases, and buying products on Internet. ES, on the other hand, deals with the deep interaction that may (or should) exist between a person and material to be learnt, and objectives similar to those existent in class, such as:

Table 1. A comparison between three sets of IHC heuristics.

Principles listed in ISO 9241-10	Heuristics from Nielsen (1992a)	Heuristics from Alsumait & Al-Osaimi (2009)
Suitability for the task	Aesthetic and minimalist design	Challenge the child, evoke child mental imagery, use multimedia representations
Self-descriptiveness	Recognition rather than recall	Learning content design (adequate vocabulary, illustrate abstract concepts)
Controllability	User control and freedom	Use appropriate hardware devices (match motor effort and children skills, prevent input errors)
Conformity with user expectations	Match between the system and the real world; error prevention	Attractive screen layout, support child curiosity, assessment (provide reports to instructor)
Error tolerance	Help users recognise, diagnose, and recover from errors	(Only lists Nilsen's heuristics)
Suitability for individualization	Flexibility and efficiency of use	(Only lists Nilsen's heuristics)
Suitability for learning (learning the software use)	Recognition rather than recall; help and documentation; consistency and standards	Design attractive screen layout (screen uncluttered, readable)

- distribute information along of space and time according to pedagogic criteria;
- intentionally leave blanks between concepts, images and words so that links be identified during the interaction with the material or even later, during a subsequent lesson;
- ask users to solve problems, analyse their reasoning and give feedback about performance;
- entice a positive affective experience on users (Immordino-Yang & Damasio, 2011).

These ES characteristics are strongly connected to the interface organization, but they clearly go beyond ergonomics and have a complex dependency on individual cognitive differences. At this point, the software designers must call into play other conceptual references, as those discussed in this article.

A promising line of research is the development of adaptive multimedia or hypermedia systems, conceived to react to user particularities; see for example Brusilovsky & Millán (2007); De Bra & Calvi (1999). Such approaches tend to emphasize the possibility to classify and model the interactions between user, the software and knowledge; they generally leave aside the complexities of human contact and classroom dynamics. In this sense, the ES is treated in a distance-learning perspective.

4. The Cognitive Theory of Multimedia Learning

The field of computer ergonomics gives support to a critic part of design: the interface and immediate (short-term) user interaction. However, the studies in this area offer limited advice concerning the teaching-learning processes that are expected to occur with the support of computers. The interaction between students and the information on the screen involves internal cognitive mechanisms that are, in principle, out of the scope of HCI.

A reference for instructional multimedia is the Cognitive Theory of Multimedia Learning (CTML) developed by the psychologist Richard E. Mayer. Undoubtedly he is best known by the investigation of the functioning of our visual and auditory channels, but he also examined other aspects of learning. This includes the effect of different sequences of presentation on retention, the influence of working with whole-part relations, or still the way information is associated and retrieved.

Mayer exploited the ability of computers to handle different information, as a means to strengthen the contact between the learner and the instructional material. A basic idea that underlies his work is the additive effect of using more than a sensory channel:

“Verbal and nonverbal systems are assumed to be functionally independent in that one system can be active without the other or both can be active in parallel. One important implication of this assumption is that verbal and nonverbal codes corresponding to the same object (e.g., pictures and their names) can have additive effects on recall” (Paivio, 1986).

It is worth pointing out that CTML present similarities with the learning styles proposed by Felder (1988) or Kolb (2005), see also (Sankey, Birch, & Gardiner, 2011), but it has roots in a physiological comprehension of the way individuals process information. The term “multimedia learning” (Mayer, 2001) refers to the process by which an individual builds a mental representation of contents presented concurrently by means of words and pictures. Each type of information undergoes a different processing in the brain. Handling textual data requires

linguistic skills to decipher a stream of symbolic data, while visual sources like a photograph, can be perceived as a whole. There is much evidence about cognitive differences associated with each type of information processing in the brain (Gardner, 1983; O'Reagan & Noë, 2001; Banich, Milham, Atchley, Cohen, Webb, & Wszalek, 2000), that support the original observations of Paivio and the subsequent work of Mayer.

CTML is based on three assumptions: verbal and visual information are treated along of different paths in the brain; each cognitive channel has a maximum processing capacity; and learning requires an active involvement of the student. These ideas are further developed into a set of principles or effects that serve to organize the design of instructional multimedia. They are listed in **Table 2** according to a categorization given by Mayer himself.

The principles listed in **Table 2** have been extensively tested (Mayer, 2008). They have been used as means to guide the design of educational materials (Park & Hannafin, 1993; Herrington & Oliver, 2000) and to predict learning results (Mayer, 2001). In order to use CTML successfully, two points should not be overlooked.

The text of the standard 9241 and the format used by Mayer to present his theory resembles a list of items. Although didactic, this can be misleading in practice, since a strict verification against a checklist can lead to a shallow evaluation (Tergan, 1998; Squires & Preece, 1999). A better perspective is that of a structured walkthrough (Yourdon, 1989); the team can inspect the product according to individual expertises, and a set of guidelines remains as a reference or a reminder. As an example, a rule that determines the positioning of buttons on the screen can be broken in favour of a different layout to present a diagram, provided that the end result will be actually advantageous to students.

Videogame developers face similar difficulties: they must pay attention to clear ergonomic directives, like limiting the number of simultaneous options on the screen; and at the same time, the project must seek the wider goal of providing a joyful experience, an extremely subjective target.

CTML is also concerned with the fact that the project of software corresponds to a fraction of the instructional design. Every element—software, books, explanations, exercises, assignments and so on, should be framed by a pedagogic structure, adapted according to the teacher style and the class needs.

5. The Theory of Meaningful Learning

The rich variety of learning theories reflects the complexity of phenomena that take place when students engage in cognitive tasks and social interaction. Works from scientists as Piaget, Bruner or Vigotski, establish what we

Table 2. CTML principles.

Principle	Category	Description
Coherence effect		The material presented to the student should avoid including information that is not part of the contents being studied.
Signalling effect		A presentation should give clues to students to guide their attention towards main points, by emphasizing or repeating information.
Redundancy effect	Reduction of extraneous load	Narrated text should not be accompanied of written text, since this can distract students from observing pictorial information.
Spatial contiguity effect		The close placement between texts and pictures reduces the effort of students to inspect material and favours learning.
Temporal contiguity effect		The presentation of verbal and non-verbal pieces of information should occur simultaneously instead of sequentially.
Segmenting effect		The presentation of information should use separable units whenever possible, instead of fusing several concepts into complex texts and pictures.
Pre-training effect	Management of essential processing	Introductory material in the beginning of a presentation, may reduce the cognitive load associated to complex information that forms the core of the learning material.
Modality effect		When pictorial and verbal information are combined, the use of narrated (spoken) text is preferable over written text.
Multimedia effect	Fostering generative processing	Explanations with text and pictures are more efficient than those presenting information using only one of these possibilities.
Personalization effect		The presentation of material should preferably make students feel part of the narration, for example using second person instead of third person conjugation.

could call “angles of approach” to deal with the task of teaching. Each system of assertions and methods forms a coherent set that can be used as the base for instructional design of books, lessons, software.

David Ausubel was a psychologist who developed research in the field of education. His work led to hypothesis concerning the functioning of the brain on an abstract, symbolic level. This view is closely related to information processing theories and cognitive approaches (Schunk, 2012).

The theory of meaningful learning was chosen in this study for two reasons. First, there is a number of links and agreements with the work of Mayer. Both authors share the viewpoint that cognitive phenomena can be explained by mechanical or information-processing metaphors, what is highly convenient in order to think about software design. Second, both theories focus the interaction between the individual and the information. In doing so, they do not preclude, neither require, adjustments to the social context. When this is necessary, the corresponding requirements elicitation must be carried out and incorporated into the project. As an example, social networking could have an effect on the sequencing of information presented on the screen. This kind of consideration falls outside the scope of this study.

Ausubel emphasized a distinction between two types of learning processes: rote memorization; and transformation of cognitive structures. In the first type of learning, individuals record facts in an arbitrary way. Examples are studying a list of items or associations, like capitals of states. In the second case new information is also stored in the brain, but this involves a network of facts that are likely to modify pre-existent knowledge. For instance, when students learn that friction is a force, their comprehension about several situations in mechanics gains new interpretations (Clement, 1982).

Learning in a meaningful way requires pre-existent knowledge that can be related to new information by the student. This process can lead to an integrative view of various concepts, or inversely to the differentiation between them. For instance, “plant” is an umbrella for things as different as “sequoias” and “roses”, while the terms “angiosperm” and “gymnosperm” can be understood as two particular types of the general concept of “plants that produce seeds”.

Not surprisingly, the anchoring mechanism does not filter or ensure the quality of information and students may acquire misconceptions (Stefani & Tsaparlis, 2009). Among the causes, teachers may propagate wrong concepts, possibly inherited in turn from their own teachers (Quílez-Pardo & Solaz-Portolés, 1995); flaws in the instructional design may cause fragmented and shallow understanding (Cooper, Grove, Underwood, & Klymkowsky, 2010); students also call into play heuristics as generalisation and simplification, which may produce distortions and false conclusions (Talanquer, 2006).

The associations between concepts can take a myriad of forms and can be represented as a network of ideas (Novak & Cañas, 2006). Some possible relations are represented in **Figure 2** and labelled according to the Theory of Meaningful Learning (TML).

In subordinate learning, new ideas are connected to the cognitive structure as particular cases of a more general concept. The super-ordinate learning works in the opposite direction, by subsuming old ideas in new higher-level concepts. Finally, in combinatorial learning information get interrelated in non-hierarchical ways. In Ausubel’s view, “meaning” is the result of the interaction between ideas. This way, by learning new ideas one can modify previous knowledge, up to the point of replacing and obliterating information.

If, during a lesson, students do not remember past knowledge or fail to recognize it as relevant, the quality of

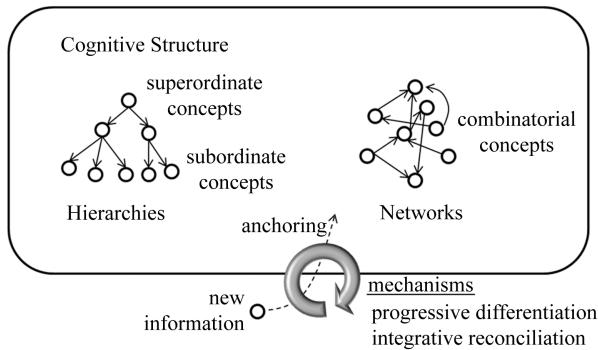


Figure 2. Overview of mechanisms from TML.

assimilation is compromised. Ausubel devised a strategy to deal with this problem, by introducing students to material that can help select and activate such memories; he called them “advance organizers”. They can take forms as diverse as pictures, texts, exercises or even spoken dialogues.

An advance organizer (AO) shows only information that is likely to be known, not new facts. It is presented at the beginning of a lesson, with the purpose of making students aware of their own knowledge. As an example, the concept of “couple” or “moment of a force” can be presented in a Physics lesson with the help of the formula $\tau = \mathbf{d}\mathbf{F}$, where \mathbf{d} and \mathbf{F} stand for the displacement and the force vector, respectively. Despite the formula being short and the mathematics uncomplicated, the abstract nature of the communication pose difficulties in class. A possible AO for this subject would be a video, showing someone struggling to remove a screw with spanners of different lengths (Koscienski, Ribeiro, & Silva, 2012).

Teachers know from experience that the design of a lesson must account for the level of knowledge of the students. However, the TML assigns a great weight to the detailed structure of such knowledge, which can explain subtle variations in tasks like comprehension of concepts and formulation of hypothesis. For instance, in the example of the moment of a force, the fact that the discussion about the formula is preceded by the presentation of a video is likely to make students more comfortable with the algebraic representation (Koscienski, Ribeiro, & Silva, 2012). If instead the formula is presented as the core of the lesson, then any examples can be perceived as a means to assimilate the mathematical notation and not as instances of application.

During the instructional design, it is possible to take advantage of the concepts of advance organizer and information anchoring even if the exact background of the students is not directly evaluated. Teachers may outline hypothetical cognitive structures that capture facts, ideas and situations that are expected to make part of the students’ repertoire. This includes things from their quotidian, contents that have been taught previously and that are revealed by dialogues in class and textual assignments. This material will serve as an organizational basis to plan the approach in class, allowing the forecast of difficulties, potential sources of misunderstandings and different angles to attack a subject. It also clarifies potential connexions between concepts.

Hypothetical cognitive structures can be represented as concept maps that, in turn, can drive the creation of storyboards (Meigs, 2003; Johnson & Scheleyer, 2003). These tools can help design the different exploration paths that students may use (Ford & Chen, 2000).

6. Connecting the Tools

Taken together, the areas discussed form a thorough basis for the whole engineering processes, during the implementation of educational software.

The discipline of HCI is directly related to software development and borrows material from psychology, but it has no specific links to education. One exception is the requirement that the interface of a program be easy to learn and understand, based on the premise that any computational tool should be unobtrusive.

On the other extreme, the theory of Ausubel dives into the domain of cognition but has no particular concern about the medium employed to deliver information. It makes little distinctions between the use of books, oral explanations and other vehicles. The emphasis is placed in higher cognitive tasks like comprehension and extraction of information.

Mayer’s work stands as a possible bridge between both worlds. The interface is treated using principles as contiguity and multimedia, while mechanisms associated with deeper assimilation are exemplified by principles as pre-training and segmentation. CTML, however, does not offer thorough answers about the construction of interfaces as does the HCI discipline, neither predicts learning outcomes with the same amplitude as TML.

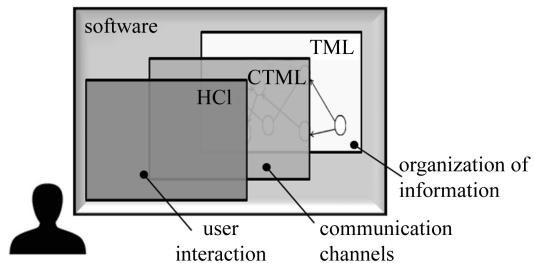
Table 3 shows a general view of the areas and their emphasis, mapped in a diagram shown in **Figure 3**. The layered structure of the diagram also reflects a temporal approach to the project, where the top level—the interface—corresponds to the last element to be refined.

TML provide the deepest level of organization. Its principles will justify the choice and the organization of the contents. It gives criteria to select metaphors, to draw relations, create exercises and activities, and any other aspects directly connected to understanding the actual meaning of the information. The contents of texts and images are produced and evaluated according to principles defined by TML, as the availability of previous knowledge.

Although teachers may be eager to draw interfaces and present ideas about how students will operate the software, such considerations should be deferred in a first moment. The team should question how the teacher

Table 3. Overview of the model.

Tool	Focus	Time scale	Intervention
HCI	Symbol recognition, workflow, short-term memory	Immediate symbol and information recognition	Immediate interaction; easiness of interface comprehension and usage
CTML	Contents assimilation	Short/long term memory	Density of information and efficiency of communicating
TML	Deep, complex learning	Life-long learning	Design of the whole instructional material

**Figure 3.** A layered view of the combination of the different tools for software design.

intends to explain each concept, idea or problem, and how that's done during a classical lesson using pencil and paper. It is only when clear strategies have been identified and understood by the team that the translation into software can take place. As a single example, during the discussion about a mathematical game showing a coordinate plan, the teacher casually mentioned that "cut the X axis" was the expression used in class to explain a certain property. That information immediately prompted the development team to change the scenery—which in principle was already validated—replacing a ball with scissors in order to strengthen the links between software, subject and verbal explanations used.

The main role of CTML is to organize the information vehicles that are assembled into the software, such as texts, videos, images. It helps to balance the use of different media and gives clues as how to combine them. In doing so, CTML may also help to consider unforeseen forms of presentation. For instance, not rarely teachers interpret the adjectives "visual" and "auditory" in a manner not compatible with CTML, and do not consider the possibility of spoken texts replacing written information on diagrams. The theory may hint teachers about potential approaches not used before, as creating a diagram adapted to the way a subject is analysed in class. Another benefit is to require an explicit control over the distribution of information. This means to quantify the number of pieces of information—diagrams, images, texts and vocabulary and to characterize their complexity. Experienced teachers may perform this task instinctively, but the same is far from being truth for programmers and graphic artists. During the project of a game, a certain pressure around the design of interface and game mechanics may build up and override other criteria for the distribution of contents.

The top level of the diagram in **Figure 3** does not simply echoes the decisions made in the previous steps. The interface is a two-way street that limits both the access of students to information and the access of the software to the way students think. For instance, a program can only make approximate predictions about the knowledge acquired, by means of tests and quizzes. Interpreting natural language is still an open problem and, as a consequence, the internal layers of the software must be projected to ensure that each new piece of information is correctly grasped and assembled into the cognitive structure. This leads back to the instructional design and point the need to introduce questions and additional explanations along of the interface, using redundancy to decrease the possibility of misunderstandings. The design of the interface, on itself, follows rules that seek to make the software transparent. Mouse clicks, keyboard actions, buttons as "ok" and "cancel", form a set of necessary objects for interaction but that should, as much as possible, be left at the edge of the attention of the student.

One last element should be considered: the affective dimension. More than a cosmetic feature, the design of characters, sceneries and storylines may have an actual influence on learning (Craig, Graesser, Sullins, & Ghosh, 2004). There is evidence that a strict separation of emotions and cognition is artificial, both from an external, behavioural view (Storbeck & Clore, 2007; Pessoa, 2008) and from observations of physiology (Phelps, 2006). The exact mechanisms are still unclear and seem to involve interdependencies between systems of beliefs,

knowledge, unconscious thoughts, feelings and other phenomena. The degree to which such relations interfere with learning and to what extent they can be controlled are subject of research, (see for instance Schunk, Pintrich, & Meece, 1996), but much remains to be explained.

For practical purposes, the sensitivity to students' preferences and cultural traits has always been acknowledged by educators. As mentioned in the introduction, it is a common practice to make adaptations in classroom, as simple as decorating it, to favour a pleasant mood. The same can be seen with the changes in style of children's textbooks and, more recently, with the use of comics and mangas in higher-education. This artistic aspect of learning materials, not only aesthetic, is rarely evoked in the literature. There is a reasonable body of research around these issues in the industry of videogames (Bethke, 2003).

A Small Scale Study

The work described in this article was originated at an attempt to circumvent the difficulties faced by K-12 Brazilian students with science contents. The specific subject was the classification of vegetables in angiosperms and gymnosperms; the primary curriculum in Brazil, as applied in classes, is quite extensive. Textbooks include a high level of detail, what comprises the identification of anatomic structures shown in photographs and diagrams, knowledge of physiology and technical names. For instance, children aged twelve may be expected to explain the difference between xylem and phloem. This emphasis on volume of data is also illustrated by the historical undervaluing of philosophy in curricula and in exams like the vestibular (university admittance test). In recent years a new trend seems to emerge, but so far textbooks remain the same.

A multimedia software was proposed as a helping tool in this context. It would give access to texts and explanations, selected images and diagrams, without the limitations found in printed material or the potential chaos of the internet. The starting idea was a sort of encyclopaedia, but during the project, the design was steered towards a game, by means of a scenario and a storyline. **Figure 4** shows one of the screens of the finished product.

The definition of the game grew out of a storyline. The plot was a conflict between good and evil, with fairies, potions and plants. Most scenes depicted a medieval castle and, although such scenery is not part of the folklore of Brazilian students, it represented nonetheless a really pleasant ambience for the public of the study. The theme was indeed selected after some input was received from children. Exercises and quizzes were included along of the software and interwoven with a background story. This ensured a continuous ambience, smoothing the rupture between "game" and "study".

In parallel to the product design, the definition of the layered approach depicted in **Figure 3** was refined. The first issue perceived in the project was the tension between different views. The teacher and the programmer had no clues about possibilities and restrictions existent in the corresponding spheres of work. This situation made



Figure 4. Screen of the implemented software (the videogame language is Portuguese).

surface again during the evaluation of the product. We will return to this issue briefly.

The initial sequencing of contents was defined by the teacher, who had twenty years of experience with the subject. Two alternatives were discussed: using the software to review the subject, or to introduce the contents for the first time to the students. In order to lower the anxiety of the teacher and also to prevent any negative outcomes, a somewhat mixed approach was selected. A quick introduction to the theme would be conducted in class, followed by the use of the computer in a subsequent lesson. This way, neither of the moments was designed to fully cover the subject; the idea that the contents should be presented repeated times was part of the overall instructional design.

The contents were laid in a sequence of units, following the scheme used by the teacher and also adopted in didactic materials as textbooks. The sequence comprised a general overview, followed by roots, stem, leaf, flower, fruit and seed. During their first walk along of the interface, the students would see these themes in this strict sequence. As the sections have been visited, they were allowed to navigate freely. This restriction helped to enforce a match between software and the lessons in class and give a sense of familiarity with the discipline.

Small texts, diagrams and photos were organized in “virtual books”, present in a library of the scenery. The students followed the plot and, when presented to an exercise, could go to the library to obtain information. The choice of texts and images was based on the indications of the teacher. In this sense the software was extremely customized. Nevertheless, the final product has been used in another school with a very different public and obtained positive reviews.

In most of the screens, the texts and images were interrelated. Both the temporal and the spatial proximity between texts and images were particularly important because of the new terms that students were supposed to learn. Despite the use of storyboards, we could later spot points where this rule has not been observed. A possible explanation for this kind of mistake was the schedule pressure, since the project was part of a master thesis. Software can be compared with this respect to the preparation of other materials, as a book text. Traditional methods of peer-revision can be used to minimize this type of problem. Larger software projects, subject to financial pressures present particularities for which specific strategies have been devised (Yourdon, 1989).

Spoken narrative was never considered in the design, because the software would be operated by couples of students sharing computers in the laboratory. Nevertheless a music track was added, with a noticeable impact on the interface.

The proposed combination of viewpoints had a truly practical impact. It supported the team providing criteria to direct discussions between programmer and teacher, the realization of studies around of sketches and helping the process of decision making. The conceptual framework acted as a filter to ideas originated from personal opinions, brainstorming or simple guessing. It also gave a higher level of confidence in order to determine the sequencing of contents distributed on the software.

Once finished, the program was analysed by a group of teachers from the areas of language, biology, arts, informatics and education. The reviews were freely conducted and the reports were positive, with a few suggestions for localised improvement. In further analysis, the relative proximity between the reviewers and the researchers may have disabled negative comments. The format chosen was open-ended questions, with the objective of letting the teachers freely express their views. This decision may also have limited the depth of analysis, because the reviewers did not have any particular background on educational software. This reaffirms the cleavage between the knowledge of computer technicians and of teachers and the need to carefully administer this aspect along of software development.

7. Final Points

The project described in this article confirmed the classic predictions of software engineering with regard to planning, schedule and project administration. The strict division of responsibilities had a positive effect on the workflow, another result that is not new but that is never overemphasized among experts of other areas, not acquainted with software production. The departure had moments of hesitation, but in a matter of three weeks the project got a life of its own and the team dynamics was stabilized.

A “pre-design” phase to assess the cognitive structures of the students improves the requirements elicitation. The team relied on the previous experience of the teacher to define the contents, but did not systematically register this knowledge; the information remained controllable thanks to the small size of the project. A printed conceptual map would have been invaluable to guide the team through development, revealing potential gaps

and acting as a reference to discuss different approaches to expose contents. We may hypothesize the development of CAD (computer aided design) tools for educational software, blending or linking concept maps into other diagrams and sources of information used in project and implementation.

Another important aspect is the complexity of designing the navigation graph of an educational application. Ausubel investigated the mechanisms underlying the storage and retrieval of networks of information, but did not address techniques for designing instructional material. Mayer studied instructional design, but with a focus on communication. Good lectures unfold complex subjects along of a clear path. It is our view that this old proven technique is still the more appropriate to lay the core design of hypermedia, as it seems to match the way we absorb information: piecewise, with clear hooks from one information to the immediately following it and, preferably, with a notion of purpose. In our study, the use of a storyline as a backbone proved to be a good solution to organize the software. According to this view, the free exploration of contents would happen backwards only; during software usage, students should follow a strict sequence. The question of self-guidance is evoked in the literature and needs to be clarified with respect to the approach proposed here.

Finally, balancing gamification with the “crude” objective of transmitting information is a problem that, to our knowledge, lacks a systematic approach. The teachers who reviewed the software missed the fact that the design underused multimedia possibilities. The non-blind nature of the review is hardly a complete explanation to the positive bias and observations about the appearance of the product. We could tentatively argue that the reviewers lacked enough familiarity with the medium in order to criticize it. However, this did not explain why they ignored points where additional concepts and contents could or should be included. In an ongoing study with a mathematical game, similar tendencies for loosing focus were observed with children, who tended to disregard instructions to focus on pure-game aspects. This seems to confirm the potential for the intentional manipulation of user perception mentioned at the origin of the text, although in this particular case it produces an undesirable, unexpected effect.

Acknowledgements

The authors would like to thank Márcia Regina Carletto for her assistance and expertise, Priscila Baer Gomes Vieira for the software programming and Welington Guimarães for the illustrations.

References

- Alsumait, A., & Al-Osaimi, A. (2009). Usability Heuristics Evaluation for Child E-Learning Applications. *Proceedings of iWAS2009*, 425-430.
- Ardito, C., Buono, P., Costabile, M. F., Lanzilotti, R., & Piccinni, A. (2009). Enabling Interactive Exploration of Cultural Heritage: An Experience of Designing Systems for Mobile Devices. *Knowledge Technology and Policy*, 22, 79-86.
<http://dx.doi.org/10.1007/s12130-009-9079-7>
- Banich, M. T., Milham, M. P., Atchley, R., Cohen, N. J., Webb, A., Wszalek, T., Kramer, A. F., Liang, Z. P., Wright, A., Shenker, J., & Magi, R. (2000). fMRI Studies of Stroop Tasks Reveal Unique Roles of Anterior and Posterior Brain Systems in Attentional Selection. *Journal of Cognition Neuroscience*, 12, 988-1000.
<http://dx.doi.org/10.1162/08989290051137521>
- Bethke, E. (2003). *Game Development and Production*. Plano, TX: Wordware Publishing, Inc.
- Brusilovsky, P., & Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web* (pp. 3-53). Berlin: Springer-Verlag.
http://dx.doi.org/10.1007/978-3-540-72079-9_1
- Clement, J. (1982). Student's Preconceptions in Introductory Mechanics. *American Journal of Physics*, 50, 66-70.
<http://dx.doi.org/10.1119/1.12989>
- Cooper, M. M., Grove, N., Underwood, S. M., & Klymkowsky, M. W. (2010). Lost in Lewis Structures: An Investigation of Student Difficulties in Developing Representational Competence. *Journal of Chemical Education*, 87, 869-874.
<http://dx.doi.org/10.1021/ed900004y>
- Craig, S. D., Graesser, A., Sullins, J., & Gholson, B. (2004). Affect and Learning: An Exploratory Look into the Role of Affect in Learning with Auto Tutor. *Journal of Educational Media*, 29, 241-250.
<http://dx.doi.org/10.1080/1358165042000283101>
- De Bra, P. M. E., & Calvi, L. (1999). AHA! An Open Adaptive Hypermedia Architecture. *New Review of Hypermedia and Multimedia*, 4, 115-139. <http://dx.doi.org/10.1080/13614569808914698>

- Felder, R. M., & Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education. *Engineering Education*, 78, 674-681.
- Flynt, J. P., & Salem, O. (2005). *Software Engineering for Game Developers*. Boston, MA: Thomson Course Technology Ptr.
- Ford, N., & Chen, S. Y. (2000). Individual Differences, Hypermedia Navigation, and Learning: An Empirical Study. *Journal of Educational Multimedia and Hypermedia*, 9, 281-311.
- Gardner, H. (1983). *Frames of Mind: The Theory of Multiple Intelligences*. Philadelphia, PA: Basic Books.
- Herrington, J., & Oliver, R. (2000). An Instructional Design Framework for Authentic Learning Environments. *Educational Technology Research and Development*, 48, 23-48. <http://dx.doi.org/10.1007/BF02319856>
- Immordino-Yang, M. H., & Damasio, A. (2011). We Feel, Therefore We Learn: The Relevance of Affective and Social Neuroscience to Education. *Mind, Brain, and Education: Implications for Educators*, 5, 115-132.
- Johnson, L. A., & Schleyer, T. K. L. (2003). Developing High-Quality Educational Software. *Journal of Dental Education*, 67, 1209-1220.
- Kirkman, B. L., & Rosen, B. (1999). Beyond Self-Management: Antecedents and Consequences of Team Empowerment. *The Academy of Management Journal*, 42, 58-74. <http://dx.doi.org/10.2307/256874>
- Kolb, A. Y., & Kolb, D. A. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. *Academy of Management Learning & Education*, 4, 193-212. <http://dx.doi.org/10.5465/AMLE.2005.17268566>
- Koscienski, A., & Soares, M. (2006). *Qualidade de Software*. São Paulo: Novatec.
- Koscienski, A., Ribeiro, R. J., & Silva, S. C. R. (2012) Short Animation Movies as Advance Organizers in Physics Teaching: A Preliminary Study. *Research in Science & Technological Education*, 30, 1-15. <http://dx.doi.org/10.1080/02635143.2012.732057>
- Mayer, R. E. (2001). *Multimedia Learning*. New York: Cambridge University Press. <http://dx.doi.org/10.1017/CBO9781139164603>
- Mayer, R. E. (2008). Applying the Science of Learning: Evidence-Based Principles for the Design of Multimedia Instruction. *American Psychologist*, 63, 760. <http://dx.doi.org/10.1037/0003-066X.63.8.760>
- Meigs, T. (2003). *Ultimate Game Design: Building Game Worlds*. Emeryville, CA: McGraw Hill/Osborne.
- Miller, G. A. (1956). The Magical Number Seven, plus or minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 101, 343-352. <http://dx.doi.org/10.1037/0033-295X.101.2.343>
- Nielsen, J. (1992a). Finding Usability Problems through Heuristic Evaluation. In P. Bauersfield, J. Bennet, & G. Lynch (Eds.), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 373-380). New York: Academic Press. <http://dx.doi.org/10.1145/142750.142834>
- Nielsen, J. (1992b). The Usability Engineering Life Cycle. *Computer*, 25, 12-22. <http://dx.doi.org/10.1109/2.121503>
- Novak, J. D., & Cañas, A. J. (2006). *The Theory Underlying Concept Maps and How to Construct Them*. Technical Report IHMC Cmap Tools, Pensacola, FL: Florida Institute for Human and Machine Cognition.
- O'Reagan, J. K., & Noë, A. (2001). A Sensorimotor Account of Vision and Visual Consciousness. *Behavioral and Brain Sciences*, 24, 939-1031. <http://dx.doi.org/10.1017/S0140525X01000115>
- Paivio, A. (1986). *Mental Representations: A Dual Coding Approach*. Oxford: Oxford University Press.
- Park, I., & Hannafin, M. J. (1993). Empirically-Based Guidelines for the Design of Interactive Multimedia. *Educational Technology Research and Development*, 41, 63-85. <http://dx.doi.org/10.1007/BF02297358>
- Pessoa, L. (2008). On the Relationship between Emotion and Cognition. *Nature Reviews Neuroscience*, 9, 148-158. <http://dx.doi.org/10.1038/nrn2317>
- Phelps, E. A. (2006). Emotion and Cognition: Insights from Studies of the Human Amygdala. *Annual Review of Psychology*, 57, 27-53. <http://dx.doi.org/10.1146/annurev.psych.56.091103.070234>
- Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction*. New York: John Wiley & Sons.
- Pressman, R., & Maxim, B. (2009). *Software Engineering: A Practitioner's Approach* (8th ed.). New York: McGraw-Hill Higher Education.
- Quílez-Pardo, J., & Solaz-Portolés, J. J. (1995). Students' and Teachers' Misapplication of Le Chatelier's Principle: Implications for the Teaching of Chemical Equilibrium. *Journal of Research in Science Teaching*, 32, 939-957. <http://dx.doi.org/10.1002/tea.3660320906>
- Rollings, A., & Morris, D. (2004). *Game Architecture and Design: A New Edition*. Indianapolis, IN: New Riders Publishing.
- Safdari, R., Dargahi, H., Shahmoradi, L., & Nejad, A. F. (2012). Comparing Four Softwares Based on ISO 9241 Part 10.

Journal of Medical Systems, 36, 2787-2793. <http://dx.doi.org/10.1007/s10916-011-9755-5>

- Sankey, M. D., Birch, D., & Gardiner, M. W. (2011) The Impact of Multiple Representations of Content Using Multimedia on Learning Outcomes across Learning Styles and Modal Preferences. *International Journal of Education and Development Using Information and Communication Technology*, 7, 18-35.
- Schunk, D. H. (2012). *Learning Theories, an Educational Perspective* (6th ed.). Boston, MA: Pearson Education Inc.
- Schunk, D. H., Meece, J., & Pintrich, P. R. (1996). *Motivation in Education: Theory, Research and Applications*. Englewood Cliffs, NJ: Merrill Prentice-Hall.
- Squirres, D., & Preece, J. (1999). Predicting Quality in Educational Software: Evaluating for Learning, Usability and the Synergy between Them. *Interacting with Computers*, 11, 467-483. [http://dx.doi.org/10.1016/S0953-5438\(98\)00063-0](http://dx.doi.org/10.1016/S0953-5438(98)00063-0)
- Stefani, C., & Tsaparlis, G. (2009). Students' Levels of Explanations, Models, and Misconceptions in Basic Quantum Chemistry: A Phenomenographic Study. *Journal of Research in Science Teaching*, 46, 520-536. <http://dx.doi.org/10.1002/tea.20279>
- Storbeck, J., & Clore, G. L. (2007). On the Interdependence of Cognition and Emotion. *Cognition and Emotion*, 21, 1212-1237. <http://dx.doi.org/10.1080/02699930701438020>
- Talanquer, V. (2006). Commonsense Chemistry: A Model for Understanding Students' Alternative Conceptions. *Journal of Chemical Education*, 83, 811. <http://dx.doi.org/10.1021/ed083p811>
- Tergan, S. O. (1998). Checklists for the Evaluation of Educational Software: Critical Review and Prospects. *Innovations in Education & Training International*, 35, 9-20. <http://dx.doi.org/10.1080/1355800980350103>
- Weinberg, G. M. (1999). *Quality Software Management. Vol. 1: Systems Thinking*. New York: Dorset House.
- Wiegers, K. (2003). *Software Requirements 2* (2nd ed.). Redmond, WA: Microsoft Press.
- Yourdon, E. (1989). *Structured Walkthroughs* (4th ed.). Upper Saddle River, NJ: Yourdon Press.

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or [Online Submission Portal](#).

