A Three-Space Design Strategy for Digital Learning Material

Jef Moonen

Introduction

Stimulated by the growth and wide availability of multimedia, the World Wide Web (the Web), and the convergence of information and communication technologies (ICT), learning materials are changing into digital learning materials, such as computer-based (available on CD-ROM) or Web-based materials. Both kinds of digital learning material are specific forms of educational software. This evolution raises the following questions: (a) Are current design and production methods for educational software still appropriate? (b) Is there a need for a new design and production strategy to cope with these new formats?

The next section offers a brief reflection on existing design methods, classifying them as structured or associative, and on characteristics of educational design. Then, the article argues that traditional structured design approaches do not appropriately address the specific needs that occur during the design-and-production process in practice. Finally, a new design approach, the Three-Space Design Strategy, is introduced. That approach, although introduced in the context of digital learning material, has a wider focus and is applicable in the broader area of social sciences.

General Design Strategies: Structured Versus Associative

General design strategies, in particular for software development (Conger, 1994), have evolved over time. They, in turn, can be seen in terms of two broad categories of general design methodologies for the social sciences. The first category is characterized by the design approach described by Simon (1969), and the second category is characterized by the design approach described by Schön (1983). The essence of the design approach advocated by Simon refers to a decomposition of the problem situation into subproblems, after which the solution of each subproblem, being downscaled in complexity, can be designed in an easier way. Characteristic of his approach is a strong end-means connection. A basic assumption, therefore, is that the problem situation is well-defined and that there is a clear agreement about the goals of the project or product. This is a systematic and well-structured approach. Decision-making based upon a logical analysis is the driving force of the activities. This design approach could be called a rational or "structured" approach.

Design methods have been traditionally based on a structured-decomposition approach. Often, however, and certainly in the social sciences, a structured-decomposition approach is not appropriate. In addition, the changing context of the situation and the (potential) influence of major stakeholders in the process will often create fuzzy situations and uncertainty. To be able to execute design processes in such situations, Schön, in contrast to Simon, suggests an approach that is based upon "reflection-in-action," whereby means and ends are used iteratively as basic elements of the situation, not assuming prior agreement about ends. Activities follow associative links within participants and interests. There is no strict predefined structure that steers the activities. Involved parties and interests jointly evolve toward a consensus about the final goal or product. This design approach could be called a relational or "associative" approach.

Depending on the specific situation at hand, both approaches are useful. As soon as (parts of) the goals of a project or product are fully agreed upon and operationalized, a structured approach is the most effective. However, when such agreement is not within reach, an associative approach is more appropriate. Given that often neither possibility describes fully the situation at hand, it seems to be a wise choice to combine, in a global design strategy, the strengths of both approaches.

Experiences from Practice

How are design strategies used in practice, in particular with respect to the design of educational software? Typically, software design methods so far have assumed a more-or-less "waterfall" approach (Humphrey, 1989) consistent with Simons' problem-decomposition conceptions. Also, most of the traditional instructional design methods are consistent with Simon's approach.

Lessons learned from past experiences with software-design methodologies for digital learning material are given by Spohrer (1998). He reports on eight years of research by the Apple Advance Technology Group, whereby more than one hundred different tools for software designing and authoring were used. He concludes from these experiences that three lessons

Jef Moonen is with the Faculty of Educational Science and Technology, the University of Twente, Enschede, The Netherlands (e-mail: moonen@edte.utwente.nl).

stand out: (a) *users first:* users had to be involved from two perspectives: cognitive fit (for usability) and social fit (for dissemination); (b) *complexity kills:* successful tools were ones that had a single key innovation that users could readily see the value of and learn to incorporate into their daily practice, especially when the innovation could be fit into an incremental improvement to an existing product; and (c) *cognitive fit is easier to attain than social fit:* the time constant in social fit is typically much longer than the time constant for cognitive fit (p. 131).

Social fit relates to the perspective of Dourish (1995), who is concerned about the static nature of the current software systems-design methodologies and argues that "the design process does not end with the delivery of a system to some community of users. Instead, it continues as they use and adapt the system" (p. 44). And, more recently, in his book about *Web Site Engineering,* Powell (1998) states that "software becomes less and less useful unless it is changed over time. Consequently, software should be designed with change in mind" (p. 37). Such ideas result in a strong tendency to relate the software design process to the context in which the finished product has to be used (Beyer & Holtzblatt, 1998; Moran, 1994). The question is: Does a structured decomposition approach to software design allow for an adequate contextual analysis, or is an associative approach more appropriate?

As an illustration to respond to this question, experiences within the Department of Educational Instrumentation (ISM) of the Faculty of Educational Science and Technology of the University of Twente are interesting. Within the ISM department, teaching and research are concentrated around the instrumental support of processes that are related to learning, communication, and information acquisition. Instrumental support is realized via media, particularly ICT, and typically results in digital learning materials. Work in this area is about the choice, design, development, implementation, and evaluation of different digital resource materials, such as computer-based, multimedia, telemedia, and Web-based learning resources. Within this department, several methodological approaches have been developed to support design and development work, each approach focussed on a specific class of a digital resource.

A typical example of such a methodology for computer-based learning material is the ISM-3 model (Diana & Kramer, 1995). The ISM-3 model describes the process of instrumentation design and development within the setting of an ISM Master's thesis assignment and can be perceived as a conceptual design-support tool. A typical Master's thesis within the department deals with the construction, implementation, and evaluation of a prototype of a digital learning product. The ISM-3 decision-support approach to realizing digital learning material, as well as other models (Collis, 1998; Collis & De Boer, 1999; Moonen, 1996), have been applied within the department for many years, serving 10–15 students each year. Based upon these experiences, three observations can be made: (a) the models do not anticipate the actual complexity of design projects and thus Master's thesis activities often extend beyond the agreed timeframe; (b) although the applied methodologies stimulate efforts to be very specific when describing the specifications of a proposed product, it turns out again and again that those specifications evolve differently during the design/development process than had been planned with the model, often as a result of the changes within the context of the work in progress; and (c) the prototype/final product is often not fully used as originally planned. Even with the support of very specific methodological guidelines based upon a structured approach and support tools, the design and production of digital learning material creates problems in terms of using the methods in practice.

In addition, design and production of educational software can often be characterized by the following attributes: (a) there is *uncertainty* about how to proceed, (b) much of the design activities have to do with *redesign,* and (c) there is a major influence of the *context.* Uncertainty is mainly caused by the complexity of the situation (many actors, many factors, unclear and almost constantly changing interactions), and the lack of comprehensive theories about the underlying processes. Design and production are often based on redesign because of the evolutionary developments in technology which often results in the need for adaptations of previously used software packages. Finally, the influence of the context in which the design has to be realized is very significant because of the many interests involved in the production of digital learning resources, in particular with respect to instructional approaches and didactical functionalities, but also because of the potential dominance of certain stakeholders in the connected decision processes. These arguments support the idea of applying a combined design approach, using appropriate elements of a structured as well as an associative perspective.

When dealing with design and development activities concerning the realization of social science concepts of an ill-defined nature—in particular in the areas of human communication, organizational issues in business and public administration, or when dealing with policy design and development—many comparable observations can be made. Also, in those areas, uncertainty about how to proceed is often the case. Many 'new' developments focus on changing and adapting an existing solution and therefore have to do with redesign. Given the many differences of opinion, the context is always of major concern and importance in a social science environment. The suggestion for a

Three-Space Design Strategy in the next section is therefore not only applicable for the realization of digital learning material, but also for the design and development of products in relation to ill-defined situations in the social sciences.

Combining Structured and Associative Approaches: The Three-Space Design Strategy

How could a combination of such global methodological approaches be worked out? The strategy followed is based upon an identification of major characteristics of the realization of digital learning material, further focus on the concerns caused by each of such characteristics, and then the choice of appropriate aspects of structured and associative design perspectives in order to overcome those problems. In this section, such a combined approach is given, followed by a description of a global strategy called a "Three-Space Design Strategy."

Fundamentals

Taking uncertainty, redesign, and importance of the context as major characteristics of current design approaches, and users first, complexity kills, and the need for social and cognitive fit as major concerns during the design and development of digital learning materials leads to the following suggestions: (a) reduce the uncertainty in a specific situation, (b) focus on reuse of already existing materials, and (c) build on input from users and create opportunities for the stakeholders to influence the design process.

Uncertainty about how to proceed with design activities is limited when dealing with the design of technical systems. When dealing with the production of a technical system, more-or-less established procedures exist, indicating the major directions of the design process. A structured design approach is possible. It would be of help if a comparable procedure could be followed for educational design activities. To reduce uncertainty in an educational design process, a kind of "temporary agreement" between the stakeholders in the projected product should be established early in the design process. Such a temporary agreement can be interpreted as a replacement of established procedures available when designing a technical system.

A temporary agreement in an educational design context can be based upon theoretical considerations, if available, but should also take into account the context and the interests of stakeholders. The temporary agreement has to be negotiated between the stakeholders and should represent a consensus between them. The consensus should be based upon what seems to be working in practice, should be global and simple, and should also incorporate a healthy amount of pragmatism. Professional practice and examples can be helpful as a basis for such a temporary agreement. Such an approach incorporates an associative design perspective and is consistent with the cognitive flexibility theory of Spiro, Feltovich, Jacobson and Coulson (1992) and with the writings of Dills and Romiszowski (1997) when the latter mention that "instructional design is, and will be, practice based on multiple paradigms" (p. xii).

In the evolution of many design schemes and strategies, especially those available for software design, there has been a tendency to incorporate the user more explicitly in the design process (Schach, 1993). Prototyping creates opportunities for such activities. Applying various cycles of prototyping during the design process also facilitates possibilities to better adjust the consensus agreement to the needs of the eventual users. Those interests were originally addressed by the stakeholders in the negotiations resulting in consensus at the beginning of the project. However, those stakeholders may have emphasized aspects which do not adequately reflect the opinions of a significant amount of potential end users. To ensure that the consensus continues to be valid after the design process has started, prototyping activities should create opportunities to check how well the agreed consensus applies to a wider variety of stakeholders and users. Such a procedure again incorporates aspects of an associative perspective to the design process. In addition, a prototyping environment creates opportunities to reuse components of already existing products. Based upon both arguments, it is recommended that a product available after each cycle of a prototyping activity has the format of a half product.

There remains the issue of cognitive and social fit. The different loops of prototyping should ensure that at the end of those processes, the final-half product should be usable (cognitive fit) by a typical end user. However, cognitive fit does not necessarily imply social fit. Indeed, an individual user could (implicitly) formulate specifications unique to his/her situation, leading to a need for a final adaptation process of the available half product. Given the specificity of that situation, only that individual end user knows the adaptations wanted, and therefore those adaptations should be done by that end user. This implies that the half product has to have incorporated possibilities, and tools, to perform such an activity. Because of the growing integration of ICT components in digital learning products, and of the flexibility incorporated in many of those components, new possibilities for adaptation activities of half products are becoming widespread.

A New Approach: A Three-Space Design Strategy

In order to deal with the possibilities described above, a new design strategy called a Three-Space

Design Strategy has been developed. Originally, that strategy was developed within the context of a multidisciplinary research project on the nature of design in the social sciences (Moonen, 1998). The application of the strategy to digital learning materials in this article is chosen as a typical example, mainly because of the obvious use of ICT in digital learning material, and the flexibility this creates.

The Three-Space Design Strategy emphasizes three kinds of activities or Activity Spaces: (a) a Consensus Space, (b) a Task Space, and (c) an Implementation Space. Within each space, the activities combine a structured and an associative approach, following main design characteristics related to the central ideas of Simon and Schön. Each space is focussed on specific activities to be performed in that space, starting from well-described inputs and resulting in well-described outputs. In the following sections, the strategy is described in more detail.

Visual Representation of the Generic Approach. The three constituting elements of the design strategy are called "activity spaces" and not "phases," to avoid the suggestion that these three elements *have* to be executed in a linear order. Within each space, the build-up is straightforward: input-process-output. Often, however, a linear order of execution between spaces (but not necessarily within spaces) will be the common practice. Therefore, each space has an input and an output, the output of the previous space being the input for the next. But it is conceivable that in a particular situation, and after the Consensus Space activities have led to the input for the Task Space, the Consensus Space continues to coexist with the Task Space, for instance through a steering committee that accompanies a project. In such a situation, the temporary agreement from the Consensus Space is not only (potentially) changing in the Task Space because of the comments received from users participating in prototyping activities, but also because the output of the Consensus Space is being reviewed by the original stakeholders or steering committee. Clearly such parallel activity will create confusion in the Task Space. The Task Space can also coexist with the Implementation Space, especially when companies release a first version of a product (in particular software products), knowing that soon after a second, third, and so on release will have to be put on the market, often to correct errors and bugs. Clearly that will confuse the end users in the Implementation Space. Figure 1 provides a visual representation of the generic approach.

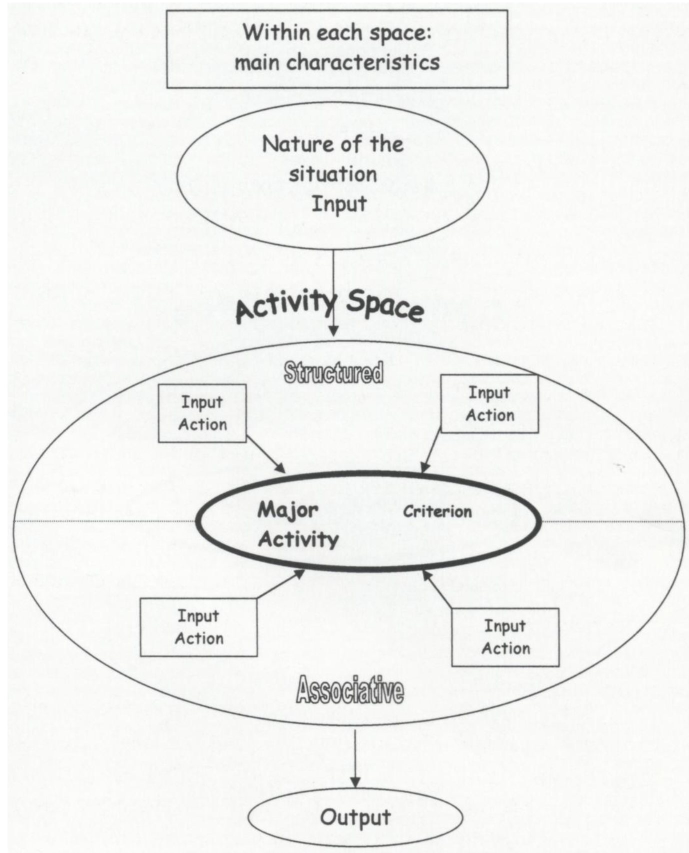

Figure 1. The generic approach for each space in the Three-Space Design Strategy.

Consensus Space. In the Consensus Space, the main objective is to move from an unstructured situation and very global ideas and specifications towards a structured design problem and functional specifications. The inputs of the process are the design problem, already-available solution ideas for the problem, and global specifications of the projected product. To support the process of reaching a temporary agreement and consensus about the product and about how to proceed, the main activity within this Consensus Space is that of balancing existing theory and decomposition of the problem (the structured perspective) with the influence of context, concerns, interests, and professional practice (the associative perspective). A process of social constructivism should take place, whereby stakeholders construct a framework, discuss, and evolve towards temporary agreement about potential solutions. Such potential solutions should merge with issues of redesign of existing resources and end up into a simplification of the problem space by the participating partners, resulting in a working consensus for the design process and product. The net results of the Consensus Space activities are initial functional specifications for the product. The management of these activities should be focussed on the process and the negotiation. Figure 2 gives a visual representation of the Consensus Space.

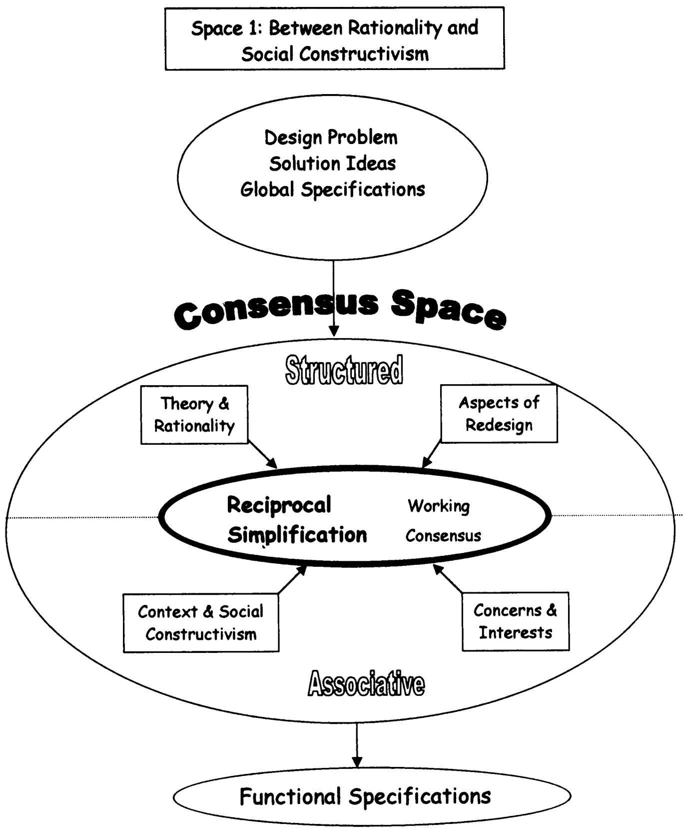

Figure 2. The Consensus space of the Three-Space Design Strategy.

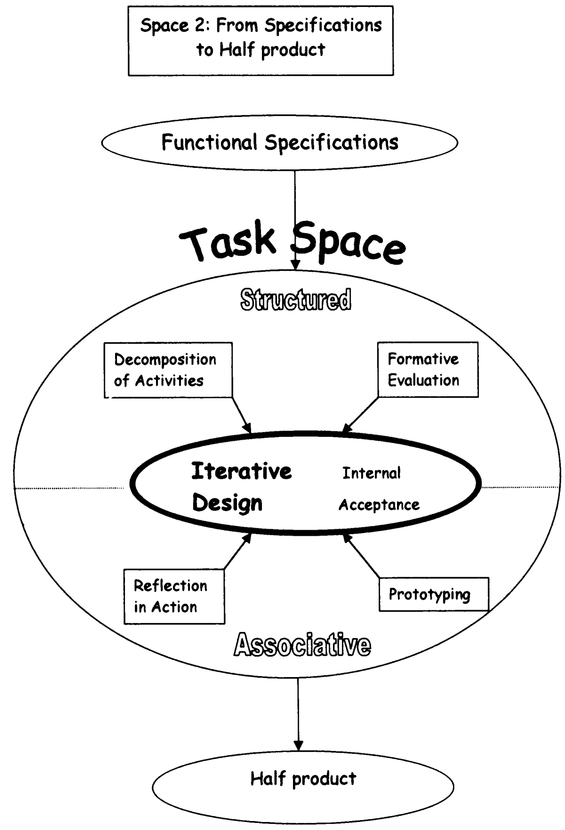

Figure 3. The Task space in the Three-Space Design Strategy.

Task Space. In the Task Space, the main objective is to iteratively formulate technical specifications and to construct, using prototyping and reusing and adapting already existing products, a series of half products. Inputs for the Task Space are the initial functional specifications that resulted from the Consensus Space. The main activity within this Task Space is a balancing activity combining decomposition of tasks based upon the functional specifications, and, through prototyping activity, formative evaluation, and reflection-in-action, confronting the partial results with the actual opinions of representatives of projected end users. This process will probably lead to an adaptation of the original functional specifications and to various versions of a half product. The 'final' half product should be "internally" acceptable, meaning that it satisfies the functional and technical specifications evolved during the Task Space. The final half product will have the format of an adaptable product. This half product will satisfy the cognitive fit requirement. The management of these activities should be product-driven, controlling milestones, deadlines and budgets. Figure 3 gives a visual representation of the Task Space.

Implementation Space. In the Implementation Space, the main objective is to confront the actual end users with the final half product, from now on called the adaptable product, and create opportunities for the end users to adapt the product to their own individualized situation and specifications. Using tools made available within available technologies (for instance, Web editors in the case of a Web resource) or tools already integrated into the adaptable product, the end users transform that product into a product that fits their specific needs. At the end of this process, the product is "externally" acceptable to the extent that the end user feels satisfied with the product. The adapted product will satisfy the social fit requirement. The management of those activities are with each individual user. Figure 4 gives a visual representation of the Implementation Space.

Internal Versus External Acceptability. In previous design methodologies based upon a structured

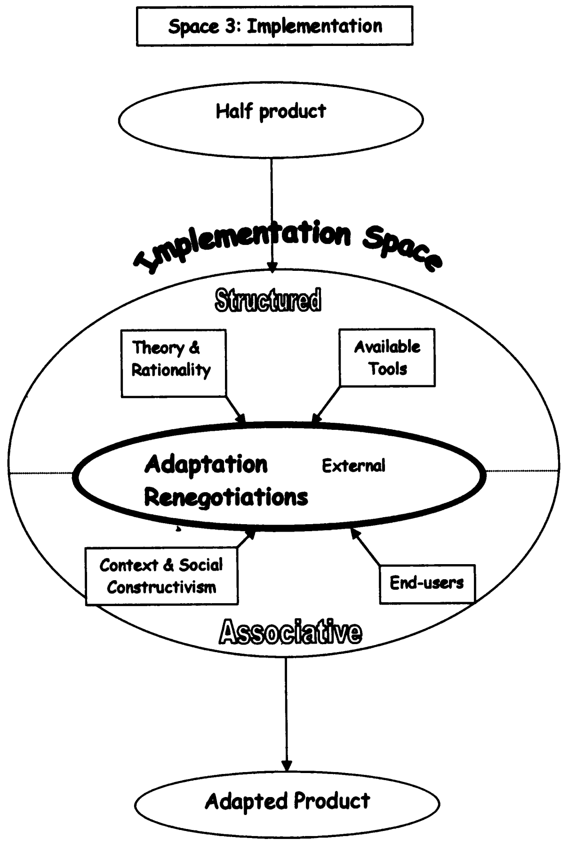

Figure 4. Implementation space in the Three-Space Design Strategy.

approach, it was implicitly assumed that when a product was internally acceptable, it would also be externally acceptable. An internally accepted product satisfied the previously agreed-upon specifications and therefore there was no further discussion about not being externally acceptable. In an ideal situation, this should indeed be the case. However, in his dissertation research about adaptable courseware, De Vries (1996) concluded that having a functional and usable product (which means that the product is made according to the requirements and functional specifications and therefore being internally accepted) does not guarantee or imply that such product will also be used by the target group, meaning being externally accepted. This situation (of high internal acceptance and low external acceptance) can happen in practice because of at least two reasons.

In the Consensus Space, when the functional specifications are formulated, these specifications are based upon a consensus reached by parties who participate in the negotiation. Often, and given realistic circumstances, end users will not be fully represented in those discussions, nor can complete contextual circumstances sufficiently be taken into account. It is therefore doubtful, but at the same time unavoidable, that the consensus reached will satisfy all potential future end users. A second reason is that there is often a time gap between the formulation of the consensus and initial functional specifications and the availability of the product. In that time period many things may have changed.

Two of the most-typical changes are: (a) the ideas that started the product design have changed, and (b) the technology that is incorporated in the product has evolved. Given these, it is reasonable to expect that, certainly in the area of digital learning materials, an internally acceptable product, if only based on initial consensus, will not necessarily be acceptable to end users (that is, externally acceptable). Consequently, it is difficult, if not impossible, to come up with "final" functional specifications as a guiding formula for a product production.

How to deal with this dilemma? A solution is to add an additional "personalized" design activity in the Implementation Space, whereby a sort of "final" specification, fitting the specific circumstances and context of that particular end user, can be (implicitly) formulated and applied. Because of the context and the individual aspects involved, that activity should preferably be done by the end users themselves.

The Main Differences Between the Three-Space Design Strategy and Traditional Design Schemes

As was said earlier, the Three-Space Design Strategy is also applicable for a much broader range of design and development activities, in particular in the social sciences, where ill-defined problem areas often occur. Two aspects differentiate the Three-Space Design Strategy from more traditional design approaches. First, there are the activities in the Consensus Space. In a traditional design scheme, a very structured needs and task analysis leads to functional design specifications. In the Consensus Space, these kind of activities are not prohibited, but on the contrary, are supplemented by an associative social constructivist approach in order to incorporate earlier experiences, specific interests, and contexts. Another crucial aspect of the Three-Space Design Strategy are the activities in the Implementation Space.

In the traditional design and development schemes, the process ends with the delivery of a product, which, in the terminology of the Three-Space Design Strategy, should be an adaptable product. The addition of the Implementation Space is mainly caused by the growing need of individualism in our society, but also because of the availability of ICT, which provides easy-to-handle and user-friendly tools to execute the adaptations of a product on an individual basis.

Examples of User-Adapted Final Products

Many software-based products have already made provisions for user-adaptation activities in the Implementation Space. For example, many of the products produced by Microsoft have features that allow the end users to adapt the product to their own personal preferences, such as, for instance, in a typical word processing package. In such a package, fonts can be changed, the layout of the screen can be changed, colors can be changed, the depth of applicability of many of the available tools (such as a spelling checker) can be adapted to individual needs, and so on. In addition, the user also has the opportunity to choose items out of many menus in order to work in an already pre-specified format (a letter format, for instance, or a report format, etc.), and to work in an environment that fits his or her particular needs. Another example is the possibility for an instructor to download preselected video segments or other resources through the Web and thus compose an individualized Web-based learning resource by integrating these segments into a Web-based course-support environment. Another example is that of instructors or students selecting modules, articles, or chapters of a book from a publisher's database and linking them together via a Web page so that in a way, they compose a book that perfectly fits their needs. This is "publishing on demand" and is already commonplace in the form of Web-based bookmark lists (Collis, 1998). Or users can construct their home-page based upon choices made available, such as in the latest my.yahoo.com Website.

Given the rapidly evolving use of the Web, designers of digital learning materials will have to incorporate opportunities and Web-based tools that will give users the potential to become the "final" designers transforming digital learning resources as adaptable products into a final, personalized product. Tools for this process will have to incorporate the need for a user-friendly adaptation process.

Conclusion

This article started by introducing a number of questions: (a) Are existing design and production methods for learning materials appropriate for practical application when dealing with new forms of digital learning material? (b) Do we need a new design and production strategy to cope with these new forms and does a Three-Space Design Strategy provide a solution? Given the developments in technologies and societal trends toward modular educational structures, it is proposed that to realize digital learning material under these new circumstances, one should follow a route that can be described in terms of a Three-Space Design Strategy, consisting of a Consensus Space, a Task Space, and an Implementation Space. In such a design strategy, negotiating, prototyping, and final adjustment of an adaptable product by the end user are main characteristics of the approach. □

References

Beyer, H., & Holtzblatt, K. (1998). *Contextual design*. San Francisco: Morgan Kaufmann.

Collis, B. (1998). *Teleware: Instrumentation of Telelearning*. Inaugural address. Enschede: University of Twente.

Collis, B., & Boer, W. De (1999). The TeleTop decision support tool. In J. van den Akker, N. Nieveen, & Tj. Plomp (Eds.), *Design methodology and developmental research in education*. Amsterdam: Kluwer.

Conger, S. A. (1994). *The new software engineering*. Belmont, CA: Wadsworth Publishing Company.

Diana, De I., & Kramer, E. (1995). ISM–3: *An approach and methodology for instrumentation research*. Manual and documentation. Enschede: University of Twente, Faculty of Educational Science and Technology; http://projects.edte.utwente.nl/cea98/cea_old/cecollab/guide.html

Dills, C., & Romizowski, A. (Eds.). (1997). *Instructional development paradigms*. Englewood Cliffs: Educational Technology Publications.

Dourish, P. (1995). Developing a reflective model of collaborative systems. *ACM Transactions on Computer-Human Interaction, 2*(1), p. 40–63.

Humphrey, W. S. (1989). *Managing the software process*. Reading, MA: Addison-Wesley.

Moran, T. P. (Ed.). (1994). Context in design. *Human-Computer Interaction, 9*(1).

Moonen, J. (1996). Prototyping as a design method. In Tj. Plomp & D. Ely (Eds.). *International encyclopedia of educational technology* (2nd ed.), (pp. 186–190). Oxford, UK: Pergamon

Moonen, J. (1998). *A design methodology for social sciences: Version 0.2*. Paper presented at the Social Science Design Conference, University of Twente.

Powell, T. A. (1998). *Web site engineering*. Upple Saddle River, NJ: Prentice-Hall.

Schach, S. R. (1993). *Software engineering*. Boston: Irwin.

Schön, D. A. (1983). *The reflective practitioner*. New York: Basic Books.

Simon, H. A. (1969). *The science of the artificial*. Cambridge, MA: The MIT Press.

Spiro, R. J., Feltovich, P.J., Jacobson, M.J., & Coulson, R.L. (1992). Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. In T. M. Duffy & D. H. Jonassen (Eds.), *Constructivism and the technology of instruction: A conversation* (pp. 23–48). Hillsdale, NJ: Lawrence Erlbaum Associates.

Spohrer, J. (1998). ATG educational research: The authoring tools thread. *SIGCHI Bulletin, 30*(2), 124–133.

Vries, Sj. de (1996). *Adaptable courseware: Design and use within secondary education* (in Dutch). Enschede: University of Twente, Faculty of Educational Science and Technology.