Adapting instructional design methods to intelligent multimedia authoring systems

Michael Baker

Centre National de la Recherche Scientifique, Laboratoire IRPEACS,
93, chemin des Mouilles, BP 167, 69131 Ecully Cedex, FRANCE.
Tel : (+33) 72.29.30.51 ; Fax : (+33) 78.33.33.70. Email : MJBAKER@FRMOP11.bitnet.

Abstract : Given the recent development of Intelligent Multimedia Authoring ("IMA") systems, there are few well-developed associated methodologies for authoring courseware. System designers have therefore turned to the automation of instructional design models, originally developed for traditional teaching. In this paper we describe one case of automating a specific instructional design model in an IMA system - the "SHIVA" system - which was based on the instructional design method of Posner and Rudnitsky. On the basis of a high-level conceptual structure of the courseware, and a set of multi-media Units of Learning Material input graphically by the author, the system automatically decides the teaching presentation sequence, whilst adapting to the students' responses. We describe results of an integrated formative evaluation program, the results of which contribute to our understanding of how to adapt existing instructional design models to the specificities of interactive intelligent multimedia and different teaching domains. In conclusion we argue that the task of authoring intelligent multi-media courseware may be viewed as a collaborative activity, involving authors and systems.

Keywords : authoring, instructional design, multimedia, artificial intelligence, training.

1. Introduction

Given the recent development of Intelligent Multimedia Authoring ("IMA") systems, there are few well-developed associated methodologies for authoring courseware. System designers have therefore turned to the adaptation and partial automation of instructional design models that were originally developed for traditional "face-to-face" teaching. However, course design processes presuppose specific *media* for the materials to be presented, and a specific *delivery system* for presenting them. The question therefore arises as to the extent to which one instructional design method, developed for given media and delivery systems (such as face to face teaching, and/or texts) needs to be modified in transferring the method to others (such as automated computer-based pedagogical decision-making and multimedia). Even in the case where this transference may reasonably be made, IMA systems may present special problems for courseware designers in understanding potentialities of new interactive media and complex pedagogical decision-making processes in AI systems.

In this paper we describe one specific attempt to incorporate a 'traditional' instructional-design method in an IMA. The system concerned is called "SHIVA", developed within the framework of the EEC DELTA Program ("Advanced Authoring Tools" project, No. D1010[1]), and the instructional design method which

[1] The research described in this paper was conducted by the following multidisciplinary consortium : CNRS-IRPEACS (France), Open University (UK), Standard Elektrik Lorenz AG (Germany), Apigraph (France) and Datamat (Italy).

it incorporates is that of Posner and Rudnitsky (1986). SHIVA allows authors to design graphically the high-level conceptual structure of courseware, and to instantiate these conceptual pedagogical objectives in multi-media units of learning material. The system dynamically determines the teaching sequence presentation as a function of the authors' decisions, the student's responses, and instructional knowledge represented in the system, using well-understood techniques in AI and Education.

The design of SHIVA thus assumes that the instructional design task may be *divided* between the authors' decisions concerning high-level course structure and the system's dynamic decisions concerning presentation order. We describe results of an integrated formative evaluation program, designed to test this and a number of other important assumptions incorporated in the design of SHIVA. The evaluation involved case-studies of teams producing pilot courseware for specific domains (geography and business English), and of experts using SHIVA to perform simple authoring tasks. Results obtained contribute to our understanding of how to adapt existing instructional design models to the specificities of interactive multimedia and different teaching domains.

The paper is structured as follows. Firstly, we describe the Posner and Rudnitsky (ibid) instructional design method. Secondly we describe SHIVA, and how it was developed in order to incorporate the instructional design method. Thirdly, we describe results of the formative evaluation programme, designed to establish the extent to which authors are able to achieve their goals with the system. Finally, we attempt to draw some general conclusions concerning automation of instructional design in IMA systems.

2. The instructional design method

The "course design" method of Posner & Rudnitsky (1986)[2] aims to bridge the gap between theory and practice in curriculum development, and is intended for secondary school teachers and teachers in training. The method needs to be described in some detail here in order to describe which parts of it were automated and which were not.

P&R make two fundamental distinctions : between curriculum and instruction, and between processes and products of planning. The curriculum (product) represents *what* is to be taught. Its development (process) involves selecting and organising *intended learning outcomes* (ILOs), guided by educational goals and values. The process of instruction is guided by an *instructional plan* (product), which defines the specific materials and instructional activities designed to achieve the ILOs of the curriculum. Once instruction is put into effect, actual learning outcomes may be assessed, and a plan for *evaluating* the course design may be defined. A simplified version of the P&R course design method is shown in figure 1.

---

[2] In this chapter all references to Posner and Rudnitsky are to Posner & Rudnitsky (1986), henceforth abbreviated to "P&R".

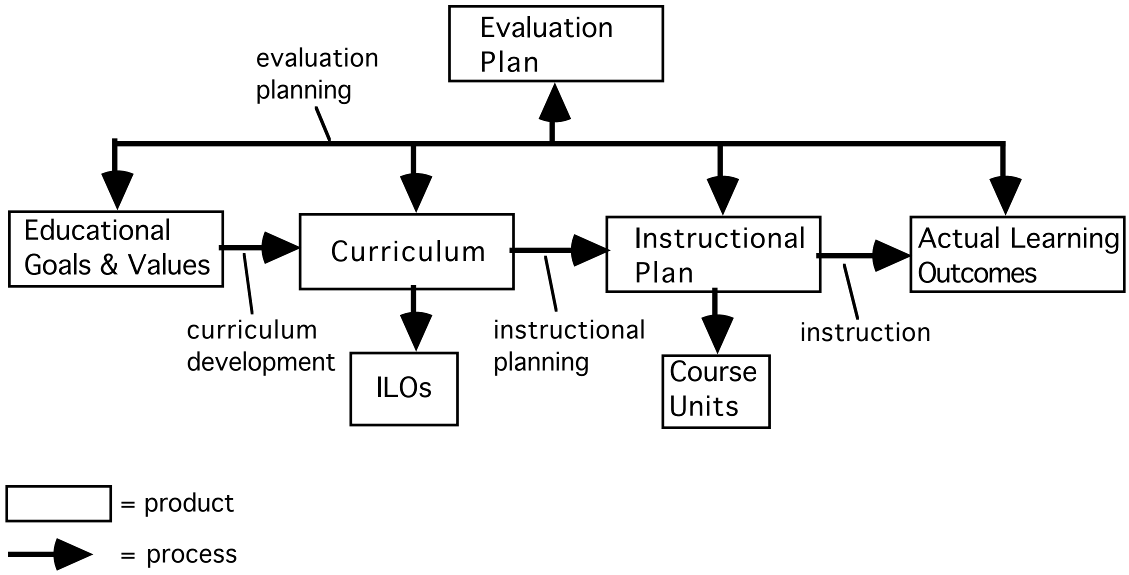


Fig. 1 The Posner & Rudnitsky Course Design Method (simplified).

Here we shall concentrate on the two main processes and products : planning the curriculum planning and planning instruction. It should be noted that although the method is presented in linear form, each new step involves revising the previous ones in the light of new information. Course design is therefore an *iterative process*.

2.1 Curriculum planning

The process of defining 'what is to be learned' is one of successive refinement, from a general course outline to specific categorised ILOs. The basic steps involved are summarised below.

1 define *course outline* (theoretical framework, initial ideas, course title)
2 select ILOs from course outline (exclude materials, activities)
3 define central questions, expand list of ILOs

4 categorise ILOs into skills and understandings
5 make conceptual maps (understandings), cognitive task flowcharts (skills)
6 analyse terminology in skills and understandings
7 arrange list of ideas and make lines between them

8 define prerequisite knowledge and students' preconceptions for ILOs
[-> revise course outline]
9 develop course rationale ("why do we have to learn this stuff ?")
[-> revise course outline]
10 categorise ILOs (cognitions, cognitive skills, psycho-motor-perceptual skills, affects)

Curriculum planning can be divided into three main phases. In the first [1-3 above] a general outline of the course is defined, and a set of ILOs is defined from it. In the second main phase [4-7 above] the current list of ILOs is categorised and structured initially (a second categorisation phase occurs later). The initial broad categorisation to be made is between "understandings" (concepts) and skills. "Understandings" are structured by making "concept maps" (essentially, semantic networks), and "skills" are structured by making flowcharts ("cognitive task analysis"). In the third phase

[8-10] the course outline and a (partially) structured set of ILOs is further refined in the light of different perspectives : the pre-existing knowledge of the target student population, the course rationale, and further more-detailed structuring of the ILOs. We shall briefly discuss each of these phases in turn, concentrating on the specific practical directions given to course designers ("how to do it")..

In the first phase the general problem is to identify possible ILOs. Advice given to course designers as to how to produce a tentative outline is as follows :

> "The best way to produce a tentative course outline is to consult several good resource books on the topic" (P&R, p. 16).

Once a draft outline is produced, the criterion for selecting ILOs from it is as follows :

> "An ILO comes into being because you think an item in your list of initial ideas or course outline is something to be learned" (P&R, p. 17).

For example, in a course on cooking specific *materials* (eg. "Rombauer's *Joy of Cooking*") and *activities* (eg. "trip to local restaurant") are therefore not ILOs. Finally, identifying "central questions" helps to give the outline *focus* : "In order to develop central questions, formulate the most important questions addressed in the course" (P&R, pp. 20-21).

In the second main phase, the list of ILOs is to be refined, structured and categorised. As to how this is to be done, P&R rely mainly on giving examples of different types of conceptual maps and skill flowcharts ; but general injunctions towards clarity, "balance", and avoidance of ambiguity and redundancy are given. In general, maps may be hierarchical or heterarchical, they may be specified in greater or lesser detail (for which several related maps may be made), and they may employ several different kinds of relationship. Once ILOs have been classified into skills and understandings, and a draft set of maps has been successively refined, ILOs may be further classified, as shown in figure 2.

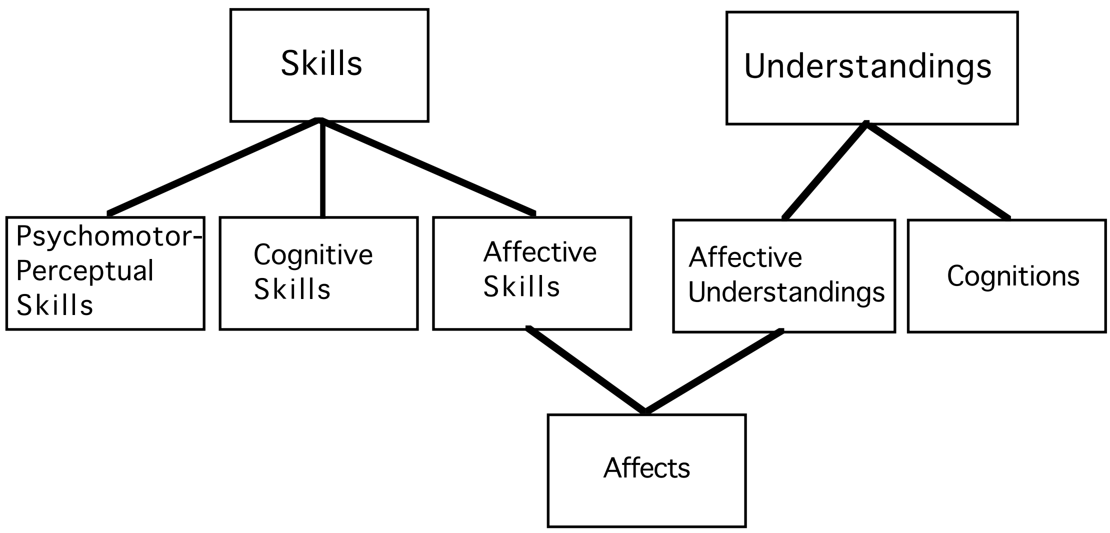


Fig. 2 Classification of ILOs (after Posner & Rudnitsky 1986)

In effect, this step requires categorisation of ILOs in terms of the type of *learning* involved.

In the third phase of curriculum planning, the categorised and structured ILOs are to be refined by considering the extent to which they may be pedagogically realised and justified, by considering the student population's prior knowledge, and the pedagogical rationale of the objectives. Given that students try to make sense of their learning experience based on what they already know or believe, P&R suggest two main practical ways in which this prior knowledge may be taken into account. Firstly, the designer should review existing courses which the students have taken. This will not, however, determine what students *know*, rather what has been *presented* to them. P&R therefore make the following suggestion :

> "The best way to find out this crucial information is to talk with them [the students] in open-ended interviews and group discussions asking them for their explanations or descriptions while listening to them very carefully. In these diagnostic settings, we can find out the extent to which their views are idiosyncratic or common to the other learners ; whether their ideas derive from prior instruction, mass media, life experience, or their own "spontaneous reasoning" … ; and possibly how deeply held their beliefs are and, thus, how difficult they will be to change."
> (P&R, p. 36).

This extract is worth quoting in full since it highlights the extent to which current automatic diagnosis in intelligent tutoring systems does not take into account the *context* of prior learning. Finally, the course rationale is "… a statement that makes explicit the values and educational goals underlying the course" (P&R p. 42). These serve to facilitate acceptance of the goals by students ("Why do we have to learn this stuff ?"), to refine ILOs (those for which a justification cannot be found should be rejected) and to guide the instruction itself (to what extent are general goals being achieved ?). It may seem surprising that the rationale is defined *after* ILOs have been defined, since "logically" the rationale should come first. However, P&R adopt the view that the rationale is best incorporated into a course design method with the function of refining an initial outline of the specific content of the course.

2.2 Instructional planning

"Instructional planning consists in planning a series of events around a particular activity, stimulus, or vehicle for communicating ideas" (P&R p. 83). The main phases are summarised below.

1. cluster ILOs into coherent and manageable course units
2. design instructional foci for each course unit
3. give a title to each unit

4. develop general teaching strategies
5. elaborate instructional foci into description of general teaching strategies
6 write rationale for each of units
7. reconsider total instructional plan

The two main stages of instructional planning involve : forming course units [1-3] and defining teaching strategies [4-7]. There is clearly a relationship between the two stages since one way in which ILOs may be 'clustered' is in terms of the ILOs relevant to a specific instructional activity (such as a visit to a museum) :

> "Instead of forming units around clusters of ILOs, units can be designed around themes and problems; around instructional activities such as projects, debates, field trips, papers, around stimuli for thinking such as case studies and photographs ; or around vehicules for communicating what we know or feel, such as books or poems."

(P&R p. 89)

In the first stage, the ILOs which are clustered together should form a coherent whole, and should thus be closely related in a concept map (in the case of understandings). Two main questions arise here : "how big should units be ?" (how many ILOs clustered together ?), and "should ILOs of different types be clustered together ? (eg. skills and understandings clustered together for a specific activity). P&R give no unequivocal answers : in general the students should be able to view the unit as coherent and manageable, and the size of such units may depend on the subject of the course. In addition to defining the units themselves, the course designer must also organise them into coherent *sequences.* For P&R, this is carried out on the same bases as those for organising the units themselves. The following are a set of different bases for clustering ILOs into units, and for organising units into sequences.

Alternative principles for organising and sequencing units :

1. world-related sequences - sequences in which there is consistency between the ordering of units and empirical relationships between events, people and objects as they exist or occur in the world (eg. sequencing of history content based on chronological sequence of events)
2. concept-related sequences - reflect organisation of the conceptual world ; consistent with way in which ideas relate to one another (eg. geometry taught deductively)
3 inquiry-related sequences - derive from the nature of the process of generating, discovering or verifying knowledge (eg. logic or methodology of physics)
4. learning-related sequences - draw on knowledge of the psychology of learning (eg empirical prerequisites ; difficulty etc. of material)
5. utilisation-related sequences - knowledge and skills organised in social, personal or career contexts.; frequency of utilisation

Clustering ILOs into coherent sequenced units may thus be viewed as a *top-down* process. However, in practical situations, teachers rarely begin 'from scratch' : they usually have some specific teaching materials, and pre-programmed activities, into which the ILOs must fit - at least to some extent - in a *bottom-up* way. They may even not be designing a course, but rather *re-designing* one. Units may therefore also be designed around themes, problems, and instructional activities such as debates, field trips, experiments, or case-studies. P&R term these "vehicles for communicating" *instructional foci.*

The second main stage of instructional planning is to design *instructional strategies* :

> "Instruction … is made up of all the purposeful activities of a teacher aimed at producing, stimulating, or facilitating learning in students. Instruction deals with how - what methods, materials, strategies, tasks, incentives, and the like can be employed to encourage learning."
> (P&R, p. 127)

"Instructional strategies" are described as specific interaction patterns between teacher and students. Choice of appropriate instructional strategy depends on the categories of ILOs taught in the unit. P&R provide an exhaustive table thus linking cognitions, cognitive skills, psychomotor skills and affects to instructional strategies. For example, in teaching cognitions, teachers should "emphasise the attributes of the concept", "ask students to find instances of the concept", "ask students to relate the concept to other concepts", "explain the implications of the assertion", etc. In teaching cognitive skills the teacher should "stimulate recall of relevant prerequisites", "furnish external prompts", "provide practice of the skill, "provide feedback", and so on. In fact, although P&R state that instructional strategies describe interaction patterns, most of their directions specify what the *teacher* should do, not the students. They do, however, briefly mention the use of simulations and rôle playing.

After the course design is finished, the course may be *evaluated* and possibily *modified*. This involves defining observable behaviours that will count as evidence that student has acquired ILOs. Information gathered in evaluation may be used to modify the course en-route. Although this stage is presented as the end of the course design process, it in fact imposes a 'backward constraint', in that the course should be designed to be *evaluable*.

3. The SHIVA intelligent multi-media authoring system

We now describe the intelligent multi-media authoring system which was based on a partial automation of the Posner & Rudnitsky course design method. Amongst those available, this method was chosen because of its relative simplicity and explicitness, which thus facilitated implementation.

3.1 Origins and development strategy

SHIVA was produced as the result of the integration of two pre-existing systems - ECAL and ORGUE - to produce an integrated prototype, which was further developed in conjunction with a formative evaluation programme. ECAL ("Extended Computer-Assisted Learning") was a system produced at the Open University, GB (Elsom-Cook & O'Malley 1989 ; O'Malley, Elsom-Cook & Ridwan 1989), within the perspective of "bridging the gap" between AI prototype ITSs and existing CAL systems used in training. The ORGUE "Courseware Engineering Tools" were produced at CNRS-IRPEACS. They comprise graphical editing tools for the production of multimedia CAL courseware. SHIVA was produced by a European consortium within the EEC DELTA programme. The partners were CNRS-IRPEACS (Lyon), Open University (GB), Apigraph (Lyon), SEL (Germany) and DATAMAT (Rome). To simplify somewhat, ORGUE and ECAL were combined in SHIVA by replacing the ECAL text-based editor with ORGUE, adding new editors, and refining the composite system during formative evaluation. Evaluation work was conducted in collaboration with the University of Nottingham, Department of Psychology (GB). We briefly describe ECAL and ORGUE before describing how they were integrated in SHIVA.

3.2 ECAL

ECAL[3] was based on a simplified version of the Posner & Rudnitsky course design model. The system was limited to teaching "cognitive skills" and "conceptual knowledge" (in P&R's terms), the only "teaching situations" incorporated are simple text presentation screens (a WYSIWYG wordprocessor is provided), and it is assumed that a curriculum has already been created. ECAL contains two main components : an authoring environment and a presentation environment. In the authoring environment, the author creates a set of frames (screens), and indexes each of them to the concept ILOs of the curriculum. This corresponds to 'clustering' the ILOs into teaching materials. The author does not explicitly represent relations between ILOs, these being calculated automatically by the system on the general basis that if ILOs are linked to the same frame, they are *related*. A simple knowledge representation is thus derived, forming one of the bases upon which the presentation system automatically determines the presentation sequence of frames. In addition to calculating the "relatedness" between ILOs, each ILO is assigned a value for its "generality" and "importance" based on the range of frames in which it occurs, and the number of frames in which it occurs. The frame presentation algorithm is also based on a simple student and dialogue model. The "numeric overlay" student model records a confidence factor (0-1) for the degree to which the student 'understands' each ILO. Thus confidence is increased after presentation of frames, and decreased or increased in the case where the student gives the correct or incorrect response to frames which pose questions to the student. The dialogue model contains a record of the current ILO on which the interaction is focussed, and a "dialogue history" (a record of which frames have already been presented, and which ILOs remain to be satisfied). At the highest level, the frame presentation algorithm selects an ILO focus (if not already determined) based on importance and generality of the ILO, or its relatedness to the previous focus. The system then attempts to choose a frame which maintains the focus from those which are linked to it. One teaching strategy used is that of "general to specific", i.e. choosing a frame with a high relevance which is more specific than the current frame, and which introduces a minimal amount of new material (student model). Since the basic presentation algorithm was retained in SHIVA we discuss it in more detail later.

3.3 ORGUE

ORGUE forms part of a general system termed a "Courseware Engineering Tool"[4], consisting of a set of graphic editors for creating multimedia presentation materials and for linking them into flowcharts which determine their pedagogical sequencing. The set of editors is specially designed to allow the author to run the courseware, in whole or in part, and thus to anticipate the learner's activity. An appropriate editor with specific funtionalities, is dedicated to each particular audiovisual task, but each editor is designed with a view to incorporating materials from others (eg. bitmap photos in graphics, sound with graphics). The editors incorporated include the following :

- MINIGR - a vectorial graphical colour editor, based on the metaphor of the overhead projector.

---

[3] ECAL has undergone further development as an independent system during and after the DELTA D1010 project. This description is therefore restricted to the version provided at the beginning of the project in 1990.

[4] The system is written in Turbo Pascal, running on PCs.

• DIESE - an editor for retrieving and inserting text into a graphical context.
• PICCOLO - a tool for modifying digitised still photographs, rather than for creating them.
• SAXO is a numerical sound editor, allowing the recording, retrieval and graphical editing of natural sound at several quality levels, in a mode compatible with the ISDN.
• DIGITISED VIDEO EDITOR - an editor for high resolution digitised video images (from a videodisc). The editor allows the insertion of video sequences into courseware, windows which are resizable during real-time playback.
• SIMENU - an editor which allows the interfaces of all the editors to be customised to authors' preferences, including the student environment.

The pedagogical flowchart interface (figure 3) allows the author to describe graphically the progressive succession of courseware. The contents of flowchart boxes may be either specific multimedia scenes, or hierarchically embedded sub-flowcharts. Links between boxes may incorporate counters (the number of times the learner has traversed that path) and boxes may also include the execution of any other Pascal program. The screen is divided into three parts. On the left, the flowcharts are created by the author, on the upper right a tree-like representation allows the author to navigate within the course structure, and on the lower right the author selects multimedia scenes to be linked into flowchart boxes from a reduced-size display. The author may run the whole or part of the course at any time (automatic compilation), and execution errors are indicated graphically on the flowchart.

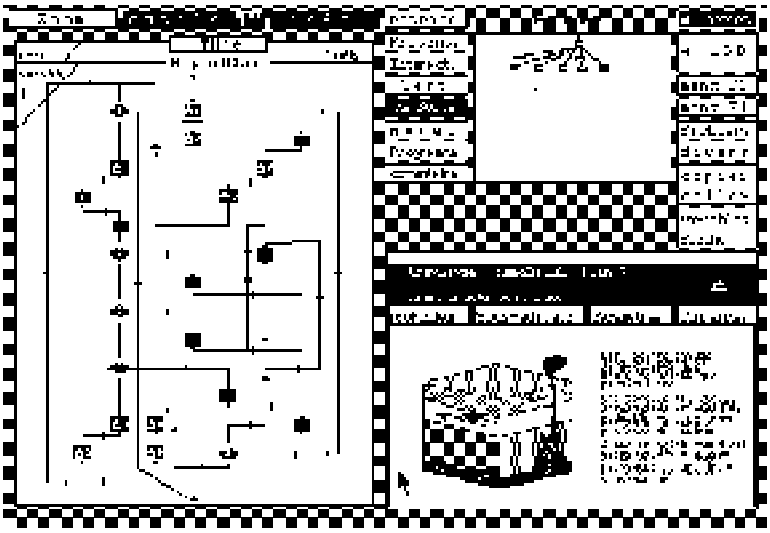


Fig.3. The ORGUE pedagogical flowchart interface

3.4 SHIVA

ECAL and ORGUE were integrated in SHIVA[5] by retaining the strongest aspects of each, and by developing new tools for authors in response to the requirements of multimedia systems and general homogeneity of the combined authoring tools. The elements of ECAL retained were the underlying conceptual knowledge representation for the curriculum (ILOs), and its attendant pedagogical decision-making program. The simple text frames of ECAL were replaced by the multimedia frames of ORGUE, with the appropriate media editors. A "Unit of Learning Material" (ULM) in SHIVA can thus contain an arbitrarily large hierarchical flowchart linking multimedia frames. The method of attaching keywords (ILOs) to text frames in ECAL was replaced by a new editor called PSAUME (figure 4).

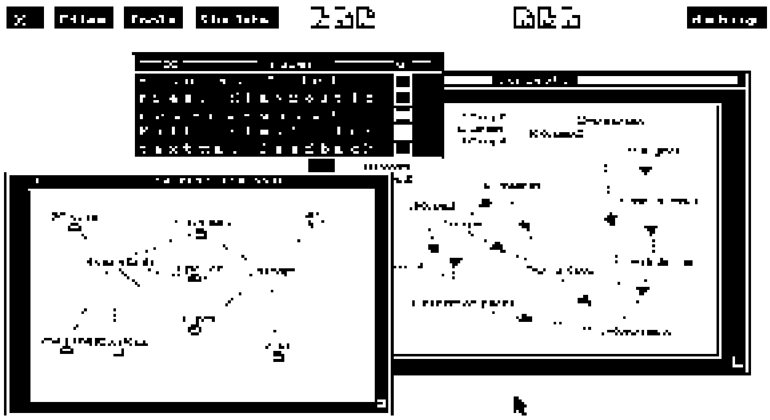


Figure 4. The PSAUME interface of SHIVA

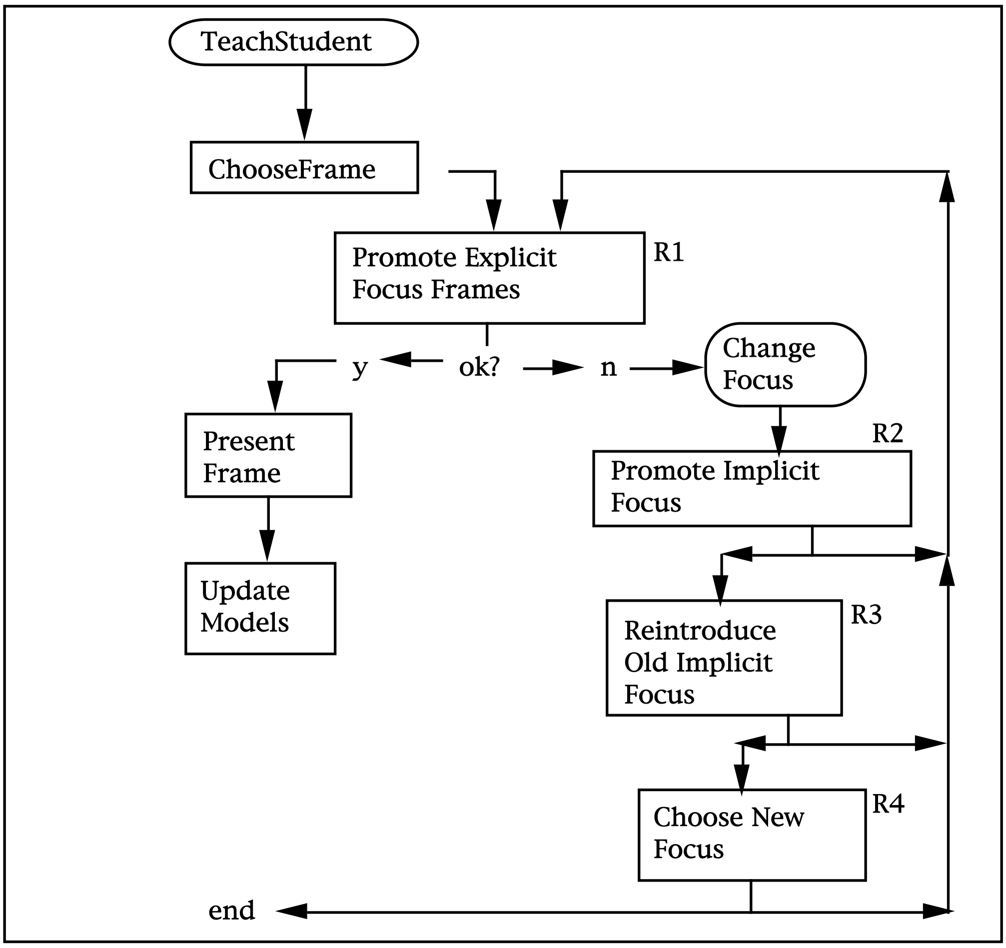
PSAUME allows authors to graphically create ILOs, and to link them to ULMs. Since ULM-ILO maps may become very complex for a large course, parts of the initial map may be represented in subviews. Although SHIVA automatically calculates relational networks between ILOs on the basis of this map, these are no longer hidden from the author, but rather may be viewed in separate CONCEPT and FRAME windows. Whilst these windows are "read only", some degree of control is given to the author over concept maps, since "pre-requisite" relations between concepts may be specified (see the right-hand window in figure 4). Just as in ORGUE, the author can inspect the contents of ULMs as represented in PSAUME.

In the development of PSAUME we began to address the problem of deciding how pedagogical decisions could reasonably be *shared* between the author and the system (in ECAL the author had only very indirect control over pedagogical sequencing, in terms of which keywords were linked to which frames). In addition to allowing authors to specify pre-requisite relations between ILOs, we therefore also attempted to allow authors to provide information on *student*

[5] A second SHIVA prototype was also produced and evaluated (SEL, Germany ; DATAMAT, Italy) in order to demonstrate that the D1010 project authoring tools can also be used to define the knowledge representation for other ITSs. This prototype integrated the "DOMINIE-II" system (implemented by the consortium of ESPRIT project 1613; see also Elsom-Cook & Spensley 1988). DOMINIE-II is designed to teach *procedural skills* involved in performing a computer-based task (such as word processing), and is thus complementary to ECAL (limited to conceptual ILOs). We shall not discuss this prototype further in the paper (see Brooks, Schmeling & Byerley 1991).

modelling, within the flowchart structure of ULMs. In part this problem arises from the fact that if ULMs are very large - and there is nothing in the system which prevents this - student modelling will necessarily be very 'coarse-grained', and in the case where students give incorrect responses within flowcharts, it will be difficult to assign 'blame' to the concept misunderstood, amonst those linked to the ULM. Authors can therefore inspect the flowchart structure of an ULM in PSAUME, and link ILOs graphically to parts of it, thus indicating "if the student passes by this route in the flowchart, decrement the confidence value of this ILO in the student model".

Although the ECAL pedagogical decision-making program was retained in SHIVA, it was adapted to include new rules relating to new activities provided for the author - for example, rules relating to concept pre-requisites. The algorithm is summarised in figure 5.



"Acceptable frame" =
not yet used & (diagnostic) only if confidence > 0 in all related concepts.
"Acceptable concept" =
not all used & not already inspected & lowest prerequisite confidence > 0.
R1: among acceptable frames, choose the one with maximum frame confidence.
R2: among acceptable concepts, choose the one with the highest relatedness.
R3: Among acceptable concepts, choose the one least discussed.
R4: among acceptable concepts, choose the one which is most important.

Figure 5. The pedagogical decision algorithm in SHIVA.

In order to determine whether the course meets their training goals, authors need to anticipate the eventual pedagogical sequence as a function of the learners' activity. In ORGUE they could do this by running the course, reacting as a hypothetical learner, and modifying the flowchart accordingly. However, since SHIVA makes automatic pedagogical decisions, authors using this system also need to understand the bases upon which these decisions are made. A set of "pedagogical debugging tools" has therefore been developed, available either from PSAUME or in the student environment. In PSAUME authors can *simulate* the course, by hypothesing students' responses to frames (either all correct, or all incorrect). The frame presentation order is displayed, together with ILO focus for each frame taught, the pedagogical rules invoked, and changes in the student model. In the student environment, changes in the student model and pedagogical rules fired may be traced as the author steps through the course.

In addition to creating new editors for ILOs, in response to the demands of ECAL, a new editor was also produced to facilitate the creation of ULMs which integrated different media. In order to facilitate 'horizontal' links between different media editors, and the creation of multimedia 'scenes', a new *multimedia editor* has therefore been developed. The editor contains a screen where windows containing scenes from each editor (video, graphics, sound, etc.) can be composed into multiple-media scenes, and provides a central point of access to each separate media editing tool.

4. A formative evaluation programme

As the name suggests, a formative evaluation is one which is carried out in order to influence the form of a system whilst it is 'taking shape'. Given the nature of the project, evaluation was concerned principally with the *authors'* point of view, rather than that of learners (although the production of a course which teaches effectively is of course the principal goal of authors).The project therefore contained an initial cycle of formative evaluation of the prototype at an early stage of development. The evaluation has had a formative role in influencing the implementation. For example, the difficulty of predicting decisions of the system gave rise to the development of simulation functionalities for authors, and results of a detailed task analysis (Payne & Green 1986), using videos of expert authors performing an experimental task, led to a number of recommendations for interface re-design.

At the outset, it was immediately apparent that two main problems needed to be addressed :

(1) the degree of homogeneity and coherence of the set of interfaces in the hybrid system ;
(2) the coherence and user-acceptance of the functional model of authoring incorporated explicitly or implicitly in the system.

These two issues are of course related since specific interfaces are provided to support authors for a large part of the authoring model. We addressed the first problem mostly using Human-Computer Interaction evaluation techniques, and the second by studies of Pilot Courseware production for specific domains. Given the aims of this paper, we describe the evaluation and its results very briefly (see DELTA D1010 Final Report for further details), in order to assess issues arising in the automation of instructional design.

4.1 Human-Computer Interaction

We used a variety of methods to assess the usability of the interfaces incorporated in SHIVA. In order to assess the consistency of the author-machine dialogue we have conducted an analysis using a modification of Task-Action Grammar (TAG) (Payne & Green, 1986) known as D-TAG (Display-oriented TAG) (Howes & Payne, 1990). D-TAG is more suitable for evaluating the interface to SHIVA since it captures the display-based nature of the interaction. Whilst it has some limitations in, for example, allowing the prediction of errors and in assessing the learnability of the interface, it has proved useful in identifying inconsistencies in task-action syntax, for example. Such methods previously been applied single interfaces, which are already largely consistent (for example, to Macdraw™). Since SHIVA is a prototype system, originating from two previous programs, it contains a large number of different interfaces, so these methods are difficult and time-consuming to apply. We therefore used more informal methods in order to focus on more restricted problems which posed genuine difficulties for authors, and within constraints of the programming effort required to modify the system. These methods for included "walkthroughs" with the system a series of prespecified tasks, by HCI-experts who were members of the SHIVA design team but who had not been involved in interface implementation.

We also conducted detailed observational studies of experienced authors using SHIVA to create and modify a small course (student contact time, approximately 30 minutes). This study enabled us to explore the extent to which authors were able to comprehend the authoring tasks required, and the way in which the system functions. Four authors were studied, in sessions lasting approximately three hours each. The authors were experienced users of ORGUE, but had never used ECAL or SHIVA before. Each subject was given one hour of training in the use of the system, including an introduction to the overall model of authoring in SHIVA, a detailed explanation of the purpose and process of creating ILO-ULM networks in PSAUME, the rules underlying the system's teaching decisions, and hands-on training in the use of the interface. The authors were seated in front of the system, with one of the evaluators seated beside them to prompt them in providing think-aloud protocols. Two video records were taken using cameras focussed on the computer screen and on the author, to record any notes made on paper or reference to documentation. Since we were only interested in evaluating the interface to ORGUE editor and PSAUME, and not the multimedia editors within this session, a set of ULMs had already been created for the purpose of the evaluation. The domain for the course was described to the authors, and they were given a list of suggested keywords with which to describe the concepts involved in the course. (This is in fact realistic, since most courseware production is done in teams, where the actual content of the course has already been specified in some way by a member of the team, and a specialist author is given the task of producing the courseware from this specification) Authors were then asked to : recreate flowcharts in ORGUE, create ILO-ULM maps in PSAUME (keywords provided), to predict the teaching sequence in the student environment, and to make some pedagogical assessment of the coherence of the teaching sequence. Where authors did not find the presentation sequence to be pedagogically 'reasonable', they were asked to modify the course using PSUAME in order to achieve their preferred outcome.

4.2 Pilot courseware production studies

We confronted the methodology of courseware design implied by SHIVA with two very different domains and learning situations. The SHIVA-Géographe project involved didactical analysis of pilot courseware, produced for teaching a specific area of geography (systematic hydrology) in a university context In the SHIVA-Westmill project pilot courseware was produced in the context of professional training closely targeted at French-speakers and to linguistic abilities required by tasks connected with the job-function of the learner. These two case studies have been carried out with partners external to the AAT project group - geography teachers at the Institut de Géographie Alpines and the Centre Informatique et Applications Pédagogiques (Joseph Fourier University, Grenoble) in the case of SHIVA-Géographe and the Westmill Business-English company (Paris) in the case of SHIVA-Westmill. These projects enabled us to identify the extent to which SHIVA could be applied to different domains, given that the system was developed with the aim of "domain independence".

The SHIVA-Géographe course emphasised teaching the dynamic aspects of the processes involved, using a *systems* approach. This approach treats knowledge from a point of view which is neither strictly declarative (or 'encyclopaedic') nor procedural, as is often the case of tutors in geography. The ULMs envisaged therefore involved simulations and the use of multiple models of the teaching domain ('water table as bath', 'as system', etc.), which poses problems for the representation of domain concepts in the PSAUME editor.

The pilot courseware produced in the SHIVA-Westmill project was designed for a commercial training rather than university teaching context. WESTMILL provide training in the use of commercial English, particularly in the Banking sector. The company has a well-defined training methodology, with a good track record of success. A specific 'audit' of training needs for each job function is provided for a particular company, in terms of a set of linguistic activities (eg "speaking on the telephone", "small talk", etc.), each of which are analysed into a set of grammatical structures required, together with the precise vocabulary which would be needed. We worked with this company in order to take a small part of one of their courses for remedial English, and redesign it in terms of the SHIVA authoring methodology. This was a good choice since the ULMs created for teaching specific grammatical tenses ("time ideas" in the WESTMILL methodology) could be reused in other courses, and adapted to other clients. This re-design resulted in a pilot production of SHIVA courseware (of approximately 30 minutes duration). The course contained : written booklets which explained the time ideas, together with short exercises, several hours of videos containing examples of time ideas with accompanying question and drill booklets using client-specific examples, audio cassettes for practising pronunciation and aural comprehension and CAL drill programs for basic tenses, produced in-house by WESTMILL.

5. Implications for automating instructional design

5.1 Automating instructional design in SHIVA

SHIVA automated only part of an existing course design model, largely because of limitations on the technological 'state of the art'. Since course design involves planning instructional situations and strategies, it is clear, however, than even given technological advance, instructional design methodologies developed for one set of situations - such as face-to-face classroom teaching - must be *transformed* to take account of new interactive teaching media. The questions arise therefore, as to which elements of instructional design transfer to intelligent multimedia environments, which new elements need to be added, and how elements transferred need to be transformed. In the case of SHIVA we can approach these questions by making the following distinctions :

Distinction 1. Elements of the P&R course design method incorporated in the SHIVA functional model of authoring are distinguished from elements which were not incorporated, and from new authoring tasks incorporated in SHIVA which are not in the P&R model ;

Distinction 2 Within the SHIVA functional model of authoring, authoring tasks for which system support is provided (interface tools, programs) are distinguished from those which are required but for which no system support is provided.

Distinction 3. Within authoring tasks for which SHIVA provides explicit support, tasks which are performed by the system are distinguished from tasks performed by the author, and tasks which are shared between author and system.

These distinctions are summarised in figure 6 below.

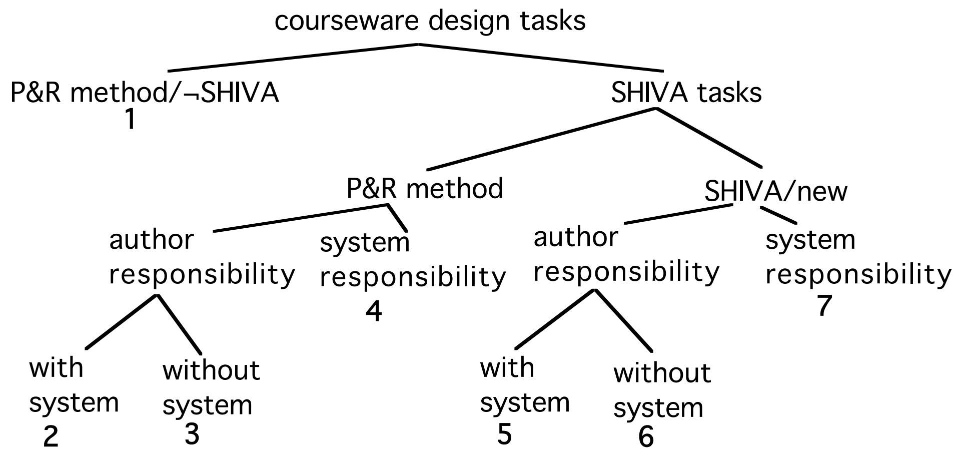


Figure 6. Comparision of course design tasks in P&R method and SHIVA

We discuss evaluation results in terms of each of these distinctions.

5.2 Comparing Posner and Rudnitsky with SHIVA

Design of courseware with SHIVA requires almost all elements of the P&R course design methodology. Although the system is restricted to concept ILOs ("cognitions"), the author must nevertheless be able to separate these out from cognitive, affective and psycho-motor skills for a given course. Identifying what counts as a "concept" ILO in fact turned out to be a major problem, as identified in pilot courseware production. In the case of SHIVA-Géographe authors had difficulty in deciding whether in the case of keywords such as "water table" these corresponded to learning objectives or not - should the student simply learn the names of objects ? In fact, this domain uses a vocabulary belonging to different lexical and language levels, corresponding on one hand to the fact that teaching takes into account the passage from the students' 'commonsense' knowledge to that of geography, and subsequently to the integration of the systemic model, and on the other hand to the hierarchical organisation of geographical knowledge.

Similar problems in authoring ILOs arose in the WESTMILL project where authors had difficulty in deciding whether an ILO was a concept or a cognitive skill. For example, authors were relutant to view grammatical tenses as concepts to be learned, since they not usually taught explicitly in their method, but rather by way of example. WESTMILL have a number of different ways of conceptualising or categorising the training domain, given its direct application in a commercial training context : they combine grammatical concepts at a number of hierarchical levels (tense, past, past continuous, etc.), specific vocabularies adapted to the particular client, groups of *competences* (such as "oral presentation", "correction of documents", "speaking on the telephone", each of which may require specific communication media), and finally, recategorisation of groups of competences, vocabularies and grammatical concepts in terms of the organisation of job functions in a specific company. In SHIVA, a single 'flat' conceptual representation is provided. Our finding was that it was extremely difficult to apply this representation to the language training domain, as conceived by the WESTMILL methodology. SHIVA thus appears to be not presently suitable for developing courses with a strong procedural component, usually taught by drill-and-practice, and where there are multiple and hierarchical ways of categorising the domain (this area is mostly covered by the DOMINIE-II approach - see footnote 5). It is rather suitable for essentially 'encyclopedic' domains, where a large quantity of factual information is to be learned. A more general point concerns the fact that SHIVA incorporates a single interaction style - the system makes decisions about which declarative knowledge should be communicated - whereas learning in most complex domains requires a combination of styles for teaching declarative knowledge, flexible control over the learning sequence, and even drill and practice of learned procedures.

A second major limitation of SHIVA in comparison with the P&R method concerns the range of specific learning materials and teaching strategies (interaction styles) incorporated. It is clear that teaching activities such as "visit to a museum", envisaged by P&R can not be directly incorporated into an intelligent multimedia learning system. However, the simple question/response interaction style of ULMs in SHIVA proved to be insufficient. In the case of SHIVA-Géographe authors conceived of the course as essentially *simulation-based* - a feature not incorporated in SHIVA - and in the WESTMILL project, facilities for incorporating client-specific texts, around which drill-and-practice exercises are based, were lacking.

A related point concerns the independence of ULMs. Since SHIVA determines pedagogical seqencing automatically, ULMs must be designed as completely independent of one another. Whilst being willing to relinquish control of frame presentation order within units, authors were reluctant to allow SHIVA the control to present units after they had established a sequence for them themselves, including the decision as to which unit must come first. Since ULMs had each to be independent of the other, this led to a lack of thematic and media coherence between units. Authors felt the strong need for an explicit explanation of the relations between successive units to learners. Although SHIVA imposes no limits on the size of ULMs, its pedagogical sequencing is most effective with as small a grain-size as possible (this permits finer-grained student modelling). However, in response to the problems stated above, authors tended to create very large ULMs in order to retain control over the teaching sequence.

There are two respects in which SHIVA incorporates functionalities which are not anticipated in the P&R model, in response to the specificities of interactive multimedia with limited AI techniques. The first concerns the provision of simulation tools for enabling authors to 'envision' interaction with learners. The second concerns the provision of a specialised editor for integrating different media

(MULTIMEDIA editor), and the design of separate editors for each media in order to facilitate the work of authoring *teams* .

5.3 System support for authoring tasks

In the standard CAL courseware design methodology (Guir 1986) conception of a course is performed on paper by a design team, which is then implemented on the machine. In part, this separation of functions was due to the necessity for programming skills in authoring languages, and the difficulty of modifying courses once implemented. One of the design goals of SHIVA was to integrate these two design phases, by supporting authors in the conception phase, and facilitating on-line modification of the course. However, a large part of the authors' task in designing courseware with SHIVA still needs to be performed off-line. In general, it is assumed that a curriculum has already been defined and that the difficult task of identifying concept ILOs has been performed. Although this process may involve creating concept maps, with different types of relations, this is not a task which the author can carry out since the concept map is calculated automatically.

An apparently 'incidental' element of the P&R method is the specification of a *pedagogical rationale* for the course, related to *educational goals*. However, this proved to be a problem for authors with respect to SHIVA, since no explanation of the rationale for its pedagogical decision-making rules was anticipated in the system. As shown from our HCI studies, authors thus found them difficult to understand and accept.

5.4 System and author responsibility for tasks

The major tasks which the system performs for the author are the creation of a relational network linking concept ILOs, and the dynamic planning of the presentation sequence as a function of it, the student model and the dialogue model. However, this raises a number of problems, related with author acceptance and understanding of the system's decisions. Authors need to understand the decision-making rules in order to be able to modify ("debug") the course so as to produce a general result which is acceptable to them. When the rules were explained to authors during the training session of our HCI studies, authors also found the pedagogical rationale for each of the rules difficult to comprehend and accept. Nevertheless we found that the authors we studied were surprisingly good at anticipating which ULM would be presented next, given the structure they had created in PSAUME. When asked to explain how they had made their predictions, authors did not use the teaching rules and decision algorithm, even though a printed summary was made available to them. They rather relied on general characteristics of the visual layout of ILO-ULM maps in PSAUME. For example, "a concept with more links to frames is more important, and so will be presented first". However, this was a relatively small course and it is not clear whether a larger course would be so predictable using such "rule-of-thumb" methods.

One way out of the dilemma of deciding whether the system or the author should carry out a task is to attempt to *share* responsibility for achieving it between author and system. We did in fact attempt to implement this principle to some extent in response to feedback from authors in the pilot studies. For example, whilst the the concept relation map is derived automatically by SHIVA, we included the possibility for authors to define pre-requisite links between concepts, and in the case of student modelling, we were driven to providing authors with the possibility of specifying concepts to be decremented within ULM flowcharts by the necessity for finer-grained student modelling in large ULMs.

6. Conclusions

Automating instructional design in intelligent tutoring systems which incorporate interactive multimedia teaching materials is a problem that has received little attention in the intelligent tutoring systems community, with some recent exceptions (Bierman, 1988 ; Pirolli & Russell, 1988 ; Spensley, 1989). The basic approach which was adopted in this research was to provide tools for supporting curriculum planning and some aspects of instructional planning (creating teaching materials to achieve curriculum objectives), leaving instructional delivery planning to the system, as a function of the students' responses. Evaluation results suggest that this way of sharing responsibility for instructional design between authors and system is not without problems, largely due to the fact that curricula are planned with a view to achieving educational goals by the means of specific teaching strategies. If authors can not understand how intended learning outcomes are to be achieved by instructional strategies , they will not be able to design courses so that the intended learning takes place. If authors do not understand the pedagogical rationale underlying dynamic instructional planning decisions, they will not be able to plan curricula in accordance with that rationale.

There are two main ways in which these requirements may be satisfied. One is by designing IMA systems which themselves understand *how* and *why* automated pedagogical decisions have been made, and which thus facilitate the generation of *explanations* for authors in terms of a coherent *pedagogical rationale*. The other is by allowing authors and systems to *collaborate* in defining the bases on which decisions are made. These alternatives are not, of course, mutually exclusive. If we look at other AI applications, such as expert systems, we find that the need for explanation facilities in expert systems has been widely accepted during the previous decade, as well as the need to implicate experts in problem-solving processes. Given the greater complexity of problems arising in the design of IMA systems, it is hardly surprising that these conclusions should be reached somewhat later in this field.

Finally, let us try to sketch what this point of view on explanation and author collaboration implies for IMA system architectures. Roschelle and Behrend define collaboration (to be distinguished from *cooperation*) as

> "... a coordinated, synchronous activity that is result of a continued attempt to construct and maintain a shared conception of a problem" (Roschelle & Behrend, in press, p. 1).

If an IMA system is to engage in *collaboration* with authors this implies that courses should be designed *incrementally*, and that both system and authors should be able to *jointly contribute* to each incremental step. This is to be distinguished from a view of an IMA system as automating and explaining some subpart of the course design process, and leaving the rest to the authors. In order to collaborate, an IMA system must possess much more higher-level knowledge of *general constraints* on course design, related to a pedagogical rationale. This knowledge should be used by the system within a process of *negotiation* (Baker 1992) with authors at each proposed course design step, rather than in making unilateral pedagogical decisions. Given that some such general constraints may be abstracted from existing course design methods (for example, coherence, generality and non-redundancy of ILOs, general size of units and coherence of their sequencing, as described earlier in our discussion of Posner and Rudnitsky's method), and current advances in generating person-machine dialogues, such IMA architectures are realistic future possibilities. Our most general conclusion, therefore, is that *instructional design should not be automated* (even partially) : it should be carried out as part of *collaborative activity between system and authors*.

Acknowledgements

This paper is based on research conducted by members of the AAT project, and was supported by the European Economic Community DELTA Program, project no. D1010.

References

1. Baker, M.J. : Towards a Cognitive Model for Negotiation in Tutorial Dialogues. In Elsom-Cook, M. & Moyse, R. (eds.) Knowledge Negotiation, Paul Chapman Publishing, London, 1992, in press.
2. Baker, M.J. : Negotiating Goals in Intelligent Tutoring Dialogues. To be published in New Directions in Intelligent Tutoring Systems, (ed.) Costa, E., Springer-Verlag, 1992, in press.
3. Barker, P. : Author Languages for CAL.MacMillan, London, 1987.
4. Bierman, D. : 'Intelligent' authoring systems: Towards better courseware. Proceedings of the Conference on Computers, Education and the Child. Urgench, USSR, 1988.
5. Brooks, P., Schmeling, A. & Byerley, P.F. : A multimedia intelligent tutoring system for Human-Computer Interaction Procedures. Human Aspects in Computing, (ed.) Bullinger, H.J., pp. 959-963. Elsevier, Amsterdam, 1991.
6. Card, S.K., Moran, T.P. &Newell, A. : The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, New Jersey, 1983.
7. DELTA Project D1010. : 'Advanced Authoring Tools', Workpackage 8, Deliverable 1: Authoring for ITS, The Open University, Great Britain. Workpackage 1, Deliverable 2: Authoring for CAL, CNRS-IRPEACS, France, 1990.
8. DELTA Project D1010. Deliverable 1. : Authoring requirements for advanced CAL, D1010, 1990.
9. DELTA Project D1010. Deliverable 2. : Authoring requirements for ITS, 1990.
10. Elsom-Cook, M. & O'Malley, C. : ECAL: Bridging the gap between CAL and ITS. CITE Report 67, I.E.T., The Open University, Great Britain, 1989.
11. Elsom-Cook, M. & Spensley, F. : Knowledge Representation in a tutoring system for procedural skills. CITE Report No. 36, IET, The Open University (GB), 1988.
12. ESPRIT Project 1613. : Evaluation of an Intelligent Tutoring System Shell for Industrial/Office Training. Deliverable 10: Final Report, 1987.
13. Guir, R. : Conception Pedagogique: Guide Methodologique. L'Association Régionale pour le Développement de l'Enseignement Multimédia Informatisé, Lyon, 1986.
14. Howes, A. & Payne, S.J. : Display-based Competence: Towards User Models for Menu-driven Interfaces. Report of Alvey/SERC Project GR/D 60355, Lancaster University (GB), British Telecom plc, 1989.
15. MacKnight, C. & Balagopalan, S. : An evaluation tool for measuring authoring system performance. Communications of the ACM, 32, 1231-1236, 1989.
16. O'Malley, C., Elsom-Cook, M. & Ridwan, E. : The ECAL Authoring Environment. CITE Report 67, I.E.T., The Open University, Great Britain, 1989.
17. Payne, S.J. & Green, T.R.G. : Task-Action Grammars: A Model of the Mental Representation of Task Languages. Human-Computer Interaction Vol.2 pp. 93-133. Lawrence Erlbaum Associates, 1986.
18. Pirolli, P. & Russell, D. : Towards theory and technology for the design of intelligent tutoring systems. Proceedings of the International Meeting on Intelligent Tutoring Systems (IMITS-88). Montréal, Canada, 1988.

19. Roschelle, J. & Behrend, S.D. : The Construction of Shared Knowledge in Collaborative Problem Solving. To appear in O'Malley, C. (ed.) Computer Supported Collaborative Learning. Springer-Verlag, in press.
20. Russell, D., Moran, T. & Jordan, D. : The instructional design environment. In J. Psotka, L. Massey & S. Mutter (Eds.) Intelligent Tutoring Systems. Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.
21. Posner, G.J. & Rudnitsky, A.N. : Course Design: A Guide to Curriculum Development for Teachers, 3rd Edition. Longman, London, 1986.
22. Spensley, F. : Generating Domain Representations for ITS. Artificial Intelligence and Education, Bierman, D., Breuker, D. & Sandberg, J. (eds.), IOS Publishing, Amsterdam, 1989.