DOCUMENT RESUME

ED 374 781 IR 016 825

AUTHOR	Boulet, Marie-Michele
TITLE	Advisor Systems: The Development Life Cycle Illustrated by Two Case Studies.
REPORT NO	ISBN-2-9802934-1-5
PUB DATE	92
NOTE	190p.; French edition available as ISBN-2-9802934-0-7.
PUB TYPE	Books (010) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC08 Plus Postage.
DESCRIPTORS	Case Studies; *Computer Software Development; Databases; *Expert Systems; Information Systems; *Intelligent Tutoring Systems; Musical Composition; Problem Solving
IDENTIFIERS	*Advisory Systems

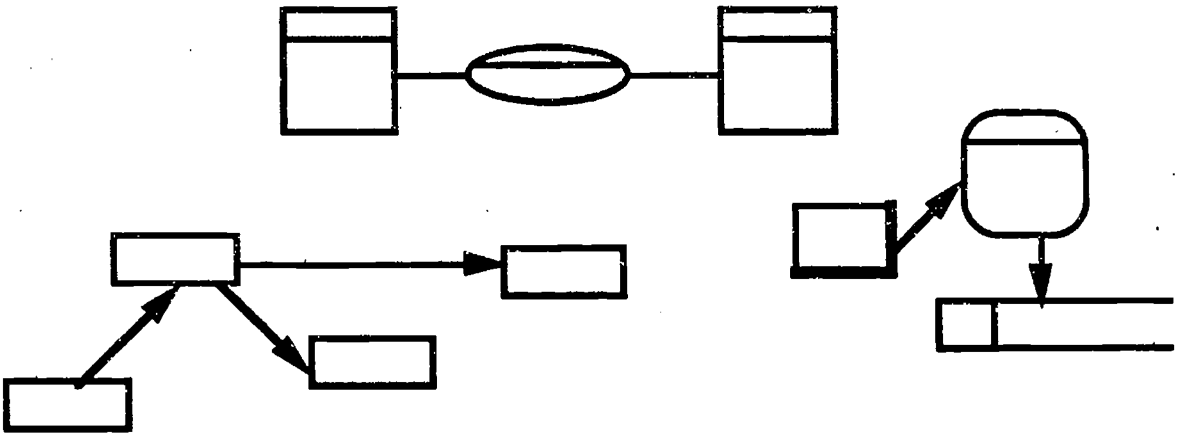
ABSTRACT

An intelligent advisor system is an adaptive system aimed at intervening when the user of any application software performs complex transfer tasks. An important aspect that needs to be taken into account when designing and developing an advisor system is the strategies and tactics used by the advisor to provide the user with an environment tailored to his or her needs and goals. This book presents the development life cycle of advisor systems as illustrated by two case studies: CODMA, an intelligent advisor in conceptual database modelling, and MUSIC, an intelligent advisor in musical composition. The introduction gives an overview of advisor systems, life cycles, and the book. For each step of the life cycle examined, the next two chapters illustrate the results obtained in the case studies. The first step, "Describing the strategy of intervention," defines the nature of knowleq, , to be recorded. The second step describes the strategy and means proposed for the knowledge acquisition. Step three describes the content, the architecture, and the implementation in the environment strategy. Finally, the fourth step describes the physical implementation of the system. (Contains 105 references.) (JLB)

MARIE-MICHÈLE BOULET, Ph.D.

ADVISOR SYSTEMS

The development life cycle illustrated by two case studies



EDITION DWAYNE

ADVISOR SYSTEMS

The development life cycle illustrated by two case studies

Version française disponible:

Systèmes conseillers: Présentation du cycle de vie et de deux études de cas.

ISBN 2-9802934-0-7

Dépôt légal (Québec and Ottawa) ii^{e} trimestre 1992
ISBN 2-9802934-1-5

MARIE-MICHÈLE BOULET, Ph.D.

ADVISOR SYSTEMS

The development life cycle illustrated by two case studies

Edition Dwayne, 1992

To Mike,

that the fortunes of life put in my road less traveled, to push me to stop, think, understand, act . . .

PREFACE

To be started, any research must be financially supported. Case studies presented here refer to researches granted by various organizations. The author wants to express her thankfulness.

The conceptual database modelling intelligent advisor research has been mainly granted by the Canadian Workplace Automation Research Center (CWARC). The author thanks the Chairperson of the center Mr. René Guindon. The authors also thanks the person responsible for the research topic to which this project was related, Mr. Joël Muzard; in the future, I hope to have other opportunities to exchange. The research has received a technical support from IBM Québec; computers lent were really used and appreciated by members of the team dedicated to the implementation. Thanks to IBM Québec for their confidence in this so risky project. Price Waterhouse (Québec) contributed to the project in terms of allocation of hours/persons; thanks to Mr. Gérard Caron who always showed a real interest in research. This research was linked to another general project aiming at developing a generic shell; the following organizations participated to the project: National Research Council of Canada (NRCC), University of Leeds, Université Laval, and Price Waterhouse.

The musical composition intelligent advisor research was granted by the Social Sciences and Humanities Research Council Canada (SSHRCC 410-90-0578). Professor Gilles Simard of the School of Music of Université Laval, was co-researcher. Thanks to the members of the comity of selection that decided to accept the project.

A research realized within a university research laboratory implies that students are hired to be part of it. They have the opportunity to work within an environment dedicated to research. The following students participated to researches described here as case studies: Luc Barbeau, Dorvalino De Melo, Francine Dufour, Guy Ennis, Steeve Jacques, Louisette Lavoie, Chantal Poirier, Josée Rochefort, Richard Roy, Sergio Slobodrian, Jacques Tang, Marc Thibault, Jean-François Tremblay. In regard of the numerous computers used within the laboratory, and because he always reacted fastly and with efficacity when it was needed, I would like to particularly thank Mr. Rodrigue Daigle from the Computer Science Department of Université Laval.

The author finalized this book while she was a visiting professor at the Department of Instructional Systems Technology of Indiana University in Bloomington. She thanks the Head of the Department, Dr. Charles M. Reigeluth. Thanks to students enrolled in my course *Advisor systems*. Their comments and questions allow me to add some precisions in regard of the nature of the tasks described in this book.

Contents

CHAPTER EIGHT

CHAPTER NINE

CHAPTER FOURTEEN

FIGURES

TABLES

CHAPTER ONE

INTRODUCTION

An intelligent advisor system is an adaptive system aiming at intervening when the user of any application software performs complex transfer tasks. According to Leshin, Pollock, and Reigeluth (1992), a transfer task: "Requires the learner to use principles to generate an appropriate procedure for any given situation" (p.51). Examples of transfer tasks are: 1- To analyze an organization and elaborate the proper conceptual database model for information needs of its managers, while using a graphic application software such as Windows™; 2- To evaluate a management information system development project in terms of information economics, taking into account both quantitative and qualitative factors, while using an estimation software such as COCOMO™; 3- To evaluate a real estate investment, taking into account both quantitative and qualitative factors, while using a spreadsheet such as LOTUS 123™; 4- To compose a melody, while using a musical writing software such as CONCERTWARE™; 5- To prepare a research proposal, while using a word processing such as WORD™.

An important aspect that must be taken into account when designing and developing an intelligent advisor is the identification of strategies and tactics that are used when the advisor proposes advice, guidance, hints, answers questions, and provides explanations. In other words, the purpose is to provide a user with an environment that is tailored to his or her needs and goals. An advisor system helps a user either to relate theory to practice, or to develop more effective problem-solving strategies, or both. By preventing inappropriate learning, an advisor keeps the user on the right track and prevents errors that could cost very much, an organization being considered.

Philosophy behind the development life cycle

An intelligent advisor is an adaptive system; adaptive is a qualitative attribute that must be defined before any other decisions related to its functions be taken. That means that the kind of transaction[1] has to be detailed before beginning to think about the various logical and physical components of the advisor. Our particular intention in this book is to consider requirements for advice-giving expert systems from a user standpoint.

The planning for advising process proposed here is user oriented. That means developers don't have in mind any a priori technical solution such as "To use Hypercard", "To develop a natural language interface", or "To use an interactive videodisc system". They are dominant ideas that tend to block out creative thinking. The approach proposed in this book is similar to the one used for developing more adapted management information systems; the underlying planning process is called strategic planning. Strategic planning relates to one type of activities that occurs during a requirements phase, being the problem analysis. The second

---

[1] According to Merrill (1988): "An instructional transaction is a dynamic interaction between the program and the student in which there is an interchange of information" (p.71).

activity is the product description. During problem analysis, analysts spend their time, through brainstorming, questioning, or both, finding out who the users really are, understanding their needs, and identifying constraints in the problem's solution. During product description, analysts describe the expected external behavior of the product to be built to solve the problem.

User oriented approach presented in this book not only considers that developers don't have in mind any a priori technical solution, but also that they don't make any a priori choice about encoding the knowledge in the advisor data structure. As example, we will refer to "black box" and "glass box model" approaches. The methodology called "issue-based tutoring" (Burton, and Brown, 1982) refers to "black box". A programmer attaches intructions to specific issues observable in the behavior of experts and of students. So, a student receives feedback about a particular behavior, when he or she chooses or fails to choose it. An example is the SOPHistical Instructional Environment (SOPHIE, Brown, Burton, and De Kleer, 1982) that uses a mathematical equation-solving process in place of the symbolic human processing. "Glass box model" refers to the use of knowledge-engineering techniques. A knowledge engineer interviews experts and, the most often, uses a rule-based formalism to design a computational representation for delivering the knowledge. This representation does not necessarily correspond to the way human reasons. In this category, we found researches aiming at turning preexisting expert systems into intelligent tutoring systems; with GUINDON, Clancey (1982) tried to implement the expert system MYCIN for diagnosing bacterial infections (Buchanan, and Shortliffe, 1984) as the expert model within an intelligent tutoring system. Other approaches to the knowledge representation are proposed in the literature. The development life cycle presented here being considered, analysts considers these numerous possibilities to decide what is the most appropriate when each help transaction has been carefully detailed that.

The user-oriented development life cycle proposed means that each decision taken in regard of each component of the advisor depends on the way developers want to provide help. Consequently, it does not aim at developing a generic advisor system. When the development of a generic advisor is the main concern, people take one or two events that happen and overgeneralize so that statements become always, every time, everybody, or everywhere; doing so, a sometimes becomes an always. In our view, developers has to select the instructional strategy they think the most useful in regard of one or many categories of users solving one or many kinds of problems related to a particular domain.

How to use the development life cycle

The development life cycle presented in this book must be considered as a suggestion, a state of mind. The author lays great stress upon this point because, in the past, she has seen too often a method blindly applied; instead of being what it was supposed to be, a mean in the authors' mind, the method became an end. Consequently, the product developed was unadapted; an unadapted product means an unsatisfied user, and that means a system not being used. In fact, we intend that the knowledge presented in this book be transferred.

As a matter of fact, transfer of knowledge was done when the two advisors described as case studies were designed and developed. For example, when the use of such a learning theory, such a method, or such a tool is recommended, that means it is recommended to adapt it. The recommendation does not mean the application of the whole method, technique, theory, or tool; it means to be inspired by it, because one or many researchers thought at length about a particular aspect of the problem to be solved, being the development of an adaptive advisor system. In the same way, we don't ask developers to fill each heading provided at each step of the life cycle. We only suggest, propose; it can happen that a heading does not concern the advisor that one wants to develop. It happens in the case studies presented. Again, the author lays great stress upon the fact that each member of the team must do transfer of knowledge.

Comments on the order of the steps

At the beginning of the process of developing any kind of computer-based instruction systems, we have used in the past approaches either based on process and data modelling, objects analysis, or rule-based formalism. Therefore, either we were looking for data and treatments of those data, objects, or rules used by experts to solve problems. In the development life cycle presented here, this way of designing such a system has been modified. The modelling of the system is done at the end of the third step. The main reason to have done this shift is the human nature.

To design and develop a system, being an intelligent advisor system claiming to be able to help a user performing a complex transfer task, is a very complex undertaking. We believe that when the main concern is to identify entities and relations, objects, processes, or rules, while abiding by the estimates of a too tight deadline, analysts are inclined to only observe the superficial aspects of the communication, to minimize its importance. They identify questions, answers, explanations, feedback from the user; moreover, they will even propose a generic system, which is a too fast jumping to conclusions. But reality is not as simple. As stated by Leshin, Pollock, and Reigeluth (1992): "... these four types of learning require great differences in instruction" (p.130). Therefore, an explanation, it can be solely an example, solely a definition, or a fixed part being a definition with a variable part being examples. It can also be sometimes a definition with or without examples, sometimes a demonstration, sometimes merely a comment on errors, sometimes a comment on error which reproduces the way the user did it, sometimes merely a message, sometimes a message linked with prerequisite knowledge ... If the reader is a little confused at this point, that was what the author aimed at. Reality is not simple, but human nature brings individuals to try to simplify it as much as they can. It is one of the greatest paradoxes that can be observed in the instructional technology industry: to propose the power of the technology to individualize, but to generalize the nature and content of transactions when designing. Analysts postulate that domain experts, being informed of their visit two days before, have in mind all the little but so important details. When the main concern is doing models, finding rules, or filling containers such as expert module, student diagnosis module, curriculum and instruction module, etc., analysts are not concern with problems of defining what is an explanation, why an explanation is formulated in such a way at a certain moment, whether explanations are always formulated in a certain way, or are sometimes formulated in that way, etc. In the other side, domain experts think that analysts are asking all the needed questions. We can imagine the result. The author insists that this way of applying development approaches does not respect the spirit of their authors. The author's observations only concern the way these approaches are used in the real world. All the philosophical foundations are the most often forgotten; producing diagrams, filling containers, in other words, focusing on the visible part is the main concern.

Considering this characteristic of the human nature, the first three steps of the development life cycle proposed here emphasize on the creative thinking process related to the identification of all the details of the strategy of intervention, the user being considered. It is in that sense that the approach is user oriented; time and again, the focus is on how the intelligent advisor will help the user. The system analysis and models associated come after. To do so, existing approaches are used, and very useful. In other words, the object of the three first steps is to assemble a detailed picture of the nature of inputs, outputs, processes, and resources required by the advisor to meet each user's needs for help; developers are not yet designing the logical or physical model of the advisor, they are preparing to design it.

Organization of the book

The general organization of the book is as follows: a step of the development life cycle being presented, the two next chapters present two different case studies aiming at illustrating the results obtained. The first case study presented in this book relates to the design and development of CODAMA, an intelligent advisor in conceptual database modelling; it is a complex case study. The second case study presented is MUSIC, an intelligent advisor in musical composition; it is far less complex than the first one.

More specifically, chapter two details tasks related to the first step, being the description of the strategy of intervention. It aims at defining the nature of knowledge to be recorded. Analysts have to define the aim of the advisor, to characterize the target users, to select the type of help transactions, to detail the main characteristics of the course of interventions, and the way the advisor adapts itself to the target users, to present the overall characteristics of the subject of interventions, and of the vocabulary of presentation, to define the knowledge depth, and to describe the interface. In other words, analysts have to detail the heart of the advisor that will be designed later. Chapter three presents the content of files issued from works done in regard of this step for CODAMA. Chapter four presents files related to MUSIC.

Chapter five describes the second step of the development life cycle. This step aims at describing the strategy and means proposed for the knowledge acquisition. The nature of knowledge having been detailed during step one, means that will be used to delimit the domain, sources that will be consulted to develop the content of interventions, and method that will be followed to do the knowledge acquisition are described. Means to evaluate the quality of sources are also proposed; it will allow the estimation of the risk of failure of the advisor. Chapter six presents the content of files issued from works done in regard of this step for CODAMA. Chapter seven presents files related to the musical composition advisor.

Chapter eight presents the third step aiming at describing the content, the architecture, and the implementation in the environment strategy. This step refers to the application of the knowledge acquisition method having been detailed during step two. In regard of the architecture, works related to the container, being the advisor structure, begin. Global characteristics of the advisor structure are modelled and documented, without taking any physical aspect into account. Tactics to ensure a successful implementation of the advisor in the environment are also described. Chapter nine presents the content of files issued from works done in regard of this step for the conceptual database modelling advisor CODAMA. Chapter ten presents files related to MUSIC.

Chapter eleven presents the fourth step of the development life cycle aiming at describing the physical implementation. Using advisor models elaborated during step three, analysts convert these logical models into physical models. Chapter twelve presents the content of files issued from works done in regard of this step for CODAMA. Chapter thirteen presents files related to MUSIC.

In regard of chapter fourteen, the function of the advisor systems technology in comparison with other tactics used in teaching/learning environment is discussed. The whole development life cycle is then presented. Further researches are finally proposed.

CHAPTER TWO

STEP 1. DESCRIBING THE STRATEGY OF INTERVENTION

To advise someone what approach to take to a given complex task, an advisor must refer to several kinds of knowledge. By definition, advise involves the making of recommendations as to a course of action by someone with actual or supposed knowledge, experience, etc. Therefore, any person or group being involved in the process of developing an advisor has to define what is the meaning of the word knowledge. That is the aim of the first step. The description of the strategy of intervention allows the identification of the nature of the knowledge to be recorded in terms of domain knowledge and user model. How this knowledge will be used is also clearly described. At any time, one has to remember that the approach is user oriented. Works done at this step focus on the identification of the boundaries of the knowledge, and of the different means of access to its various parts. In other words, the advisor will provide explanations in such a way, depending on the strategy of intervention selected; this strategy of intervention being detailed, it will allow the identification of the domain knowledge, and of the knowledge of the user to be recorded.

Aim of the advisor

At the beginning of the process, the aim of the advisor is formulated in terms of what it will bring to the user.

Examples are: Does the advisor aim at furthering the knowledge transfer? Does the advisor aim at sharing expertise acquired by only some members of an organization? Does the advisor aim at helping a user to perform some specific tasks? Does the advisor aim at providing a flexible access to training material? Does the advisor aim at ensuring that all the members of an organization perform a task in a same way?

Target users of the advisor

The target users of the advisor are characterized. That corresponds to answer to the following question: What is (are) the category(ies) of users? For example, are the users, individuals who have only a weak knowledge of the domain? Are the users, individuals who have a satisfactory level of knowledge in view of concepts, principles and methods, but having had no or few opportunities to transfer their knowledge? Are the users, individuals who have a satisfactory level of knowledge in view of concepts, principles and methods, and having had in the past opportunities to transfer their knowledge, but having been appointed to another position for a certain period of time? An example of advisor developed in regard of many categories of users is presented at chapter three. Chapter four presents an advisor developed in regard of a single category of users.

Type of help transactions

What type of help transaction(s), characterized in term of resource, will be provided to the user is selected: 1- An active resource, i.e., an advisor that monitors the user progress in performing tasks, and analyzes errors, so that it can suggest a better method; 2- a passive resource, i.e., an advisor responding to requests for help and questions from the user; 3- a pedagogical resource that includes a teaching program designed to meet the needs of occasional users, or users having a poor understanding of certain concepts, methods, principles, etc., 4- a mixed resource. The case study of the chapter three is an example of a mixed resource. The case study presented at chapter four describes a assive resource.

Main characteristics of the course of interventions

For each resource, the main characteristics of the course of interventions are described. For example, a passive resource which, when the user recognizes the need for some pieces of information and questions the advisor, replies to a question related to a precise aspect, does some comments on the task the user is performing, makes up examples of the task, and does comments on those examples. In that case, the main characteristic is to remind the user of concepts, principles, and rules, that have seemingly been forgotten.

How the advisor adapts itself to the user

It was previously stated that the approach is user oriented. That means that strategies and tactics used by the advisor are planned in such a way that it will be able to adapt itself to the user. This description has to be done for each resource selected. It is to identify the attributes of an effective individualized intervention, and to describe how the individualization will be ensured. In the next paragraphs, several possibilities are presented. This list is not intended to be complete; it represents examples. Also note that it can be decided to mix together many possibilities.

To ensure individualization, it can be decided that the advisor will engage the user in a dialog; according to the content of the user feedback, the advisor will modify the content of its next interventions. More details have to be provided. Will the advisor engage the user in a dialog each time it does an intervention, and regarding each feedback, modify the content of the intervention that follows? Will the advisor engage the user in a dialog each time it does an intervention, and regarding this unique feedback, modify the content of the intervention that follows, and also record a set of feedbacks in order to be able to provide later a more global intervention taking into account this set of data? Will the advisor engage the user in a dialog, record a set of feedbacks before providing an intervention taking into account this set of data? Answering these questions allows the identification of the user knowledge that the advisor should have; the main characteristics of the user model are defined. To be able to tailor the intervention to the user current knowledge and skills, an advisor must record and analyze data with regard to this user. The description of the attributes of an effective individualized intervention brings the user model to be described, and how and when this model is updated.

Instead of engaging the user in a dialog, it can be decided to grade him or her. The advisor can propose levels; the main concern is to determine the number of levels. When only few levels are proposed, such as beginner, intermediate, and expert, they may not reflect the current status of the user knowledge; that means the advisor won't be able to give real individualized explanations. On the other hand, when many numbers of levels are proposed, it becomes very difficult for any individual to grade himself or herself. Moreover, many researches show that an individual is not really able to evaluate his or her mastery of a domain (Nisbett, and Wilson, 1977; Rich, 1983). But this approach has been used in the past, and has been proved to be helpful when combined with other approaches; consequently, it must not be totally rejected.

Another way to determine the user knowledge level is to use a grading test; because it can lead to irritations of the user, the process must be carefully planned and managed.

Making inferences in regard of the user knowledge level is another possibility. The nature of user requests can be analyzed. As "a transfer task consists of principles and decision rules that an expert uses to generate the appropriate performance for any given situation. It also consists of secondary content, such as concepts and information" (Leshin, Pollock, and Reigeluth, 1992, p.92), this analysis can be done in regard of establishing to what extent the secondary content is mastered. That corresponds to answer to the following question: Do the user requests refer to prerequisite capabilities corresponding to the lower level of a hierarchy of intellectual skills? Hill and Miller (1988) found that 61% of the users' requests for help relates to a specific objective. McCoy (1983) observed that users wanted informations about intermediate objectives, and about the sequence of actions to be performed to obtain a correct result. These observations bring out the advantage of a knowledge representation made in regard of prerequisites. The use of a model that adheres to the cumulative learning hypothesis when the analysis of content is done, can facilitate the identification of the prerequisites in regard of a specific complex transfer task. This prerequisite knowledge being identified, it is possible to link certain requests to a level within a hierarchy; this level can after be used to determine the user one. The work done by Gagné and his collaborators is an example of such a model.

Gagné and his collaborators state that the most important condition for ascertaining whether an individual has learned something is whether or not the individual has learned the prerequisites (Gagné, 1968, 1974; Gagné, and Brown, 1961; Gagné, Mayor, Garstens, and Paradise, 1962). They mention that each person approaches a task with a different set of previously acquired capabilities; it is very important to teach the prerequisite capabilities to an intellectual skill that the person has not yet mastered, and to enable the person to review them at any time. To identify the prerequisite knowledge, Ausubel's works (1968) related to links between propositions and the model proposed by Gagné (1977) can be used. It will allow the division of an intellectual skill into objectives that will then be categorized.

To break an intellectual skill into objectives that correspond to prerequisites, the following question has to be answered: What must an individual know before being able to perform task X? The resulting structure has at its top, the primary task to be performed. Several branches may originate from the top towards knowledge and skills considered fundamental to the task execution. These in turn become tops in a recursive process.

Answering questions such as "Define conceptual database model", or "Define tonality", is associated to the mastery of a capacity named idea. An idea refers to a set of common characteristics that several actions or objects have; it allows an individual who possesses a mental representation of these actions or objects to be able to communicate these representations to other individuals (Brien, 1980). Ideas are at the lowest level of a hierarchy. Answering questions such as "State the philosophical foundation of conceptual database modelling", or "Name the main elements of a conceptual database model", refers to a capacity named proposition. A proposition is made of several linked ideas allowing an individual to have a mental representation of events (visible or not), and to be able to communicate his or her representation of events (Brien, 1980). The hierarchical representation being considered, the capacity named proposition is one level above idea. At the level above proposition, we found the interrelated sets of propositions. They are made of several linked propositions. Their mastery allows an individual to be able to answer questions like "Explain where properties of a conceptual database model can be found" or "Explain the musical composition process called question-answer". To be able to answer those questions, a person must refer to many ideas and many propositions, and integrate all those ideas and propositions in a whole. That does not mean the individual is able to perform the task; he or she is only able to explain how to perform it (Brien, 1980). Idea, proposition and interrelated set of propositions relate to a category named verbal information, which is prerequisite to capacities related to another category named intellectual skill.

There are four subcategories of intellectual skills: discrimination, concept, rule, and higher-order rule (problem solving; Gagné, 1977). Discrimination is a capacity allowing the recognition of differences between things (visible or not), without being able to put a label on these differences. Assuming someone is looking at a conceptual database model, distinguish rectangles from ellipses is an example of discrimination; however, this person is not able to associate the name entity to each rectangle, and the name relationship to each ellipse. Discrimination is one level above verbal informations in the hierarchy. Concepts are at the next one. There are two classes of concepts: concrete and definite. Concrete concepts are objects or events classes. When someone, aiming at elaborating a conceptual database model, is studying the forms used within an organization, he or she is able to associate the name property to some headings. When it is not possible to perceive characteristics, as for the identifier of a relationship (conceptual database modelling domain), the capacity is named definite concept. At one level above concept, we have rules. It is a procedure allowing to change a given situation A in a wanted situation B (Brien, 1980). An example is the capacity to determine the correct identifiers for each entity of a conceptual database model. At the last level of the hierarchy, we find higher-order rules (problem solving). They are more complex than rules. Facing a new context when elaborating a conceptual database model, to set out the proper cardinalities is an example of higher-order rule. Note that many other models exist. Ausubel (1960) proposes advance organizers, Bruner (1962, 1963, 1966a, 1966b) proposes a spiral curriculum, Landa (1976), an approach based on algorithm, and Reigeluth (1987; Reigeluth, and Stein, 1983), the elaboration theory. The Gagné's model has been used as the basis allowing to infer the user knowledge level for CODAMA; the Ausubel's advance organizer model is used to produce remedial teaching. The Bruner's spiral curriculum has been used to design MUSIC.

To infer the user knowledge level, the advisor can refer to a list of systematic errors. One must be able to produce such a list of errors with their causes; here again, models exist. We refer to the field of measurement and evaluation in education (Scallon, 1988a, 1988b). When a list of systematic errors is used, the division of the domain is not the same as what has been described above regarding the various levels of a hierarchy. Consequently, one can decide to divide the domain in terms of systematic errors and to link a question or a list to each error. In the same way, one can decide to divide the domain according to a prerequisite structure and to link systematic errors to an objective defined according to the hierarchy. An example of application of this last possibility is presented in CODAMA case study; the models used are the multiple facet scheme proposed by Guttman and Schlesinger

(1967a, 1967b), and the hierarchical test (Scallon, 1988a, 1988b).

The user knowledge level can also be inferred in regard of the frequency of the requests for help relatively to a same topic. The domain might be divided according to topics or prerequisites. The main difficulty is to define the norm: how many requests are necessary to infer that it is enough. Here again, learning theories are very useful. Think about the success level of 80% proposed by mastery learning theory researchers (Anderson, and Block, 1977; Block, and Anderson, 1975; Bloom, Hastings, and Madaus, 1971). This idea was used when designing CODAMA.

The user knowledge level can be inferred in regard of the frequency of the errors made by a user for a same topic. Here again, the main problem is to determine the norm. Each error can be associated to a hierarchy of prerequisite capabilities or to a division of the domain in terms of topics. This way of making inferences is used by CODAMA.

Instead of taking decisions in regard of a level, it can be decided to manage the organization of the content. For example, the spiral curriculum model proposed by Bruner (1962, 1963, 1966a, 1966b) can be used to produce interventions in regard of a progressive integration of several prerequisites. The spiral curriculum model is used in the musical composition advisor case study.

Table 2.1 summarizes possibilities discussed here.

Table 2.1

Summary of possibilities discussed in regard of how the advisor adapts itself to the user

For all types of resources			
How does the advisor adapt itself to the user?	->dialog with user	->unique feedback	
		->unique feedback & set of feedbacks	
		->set of feedbacks	
	->level	->user decides	
		->test of classification	
		->inference	->nature of questions asked
			->systematic errors
			->frequency of requests for help
			>frequency of errors
	->content	->topics	
		->hierarchy of prerequisite capabilities	
		->spiral	
		->elaboration	
		->algorithm	
		->advance organizer	

Keeping in mind the user oriented approach, and assuming that some weaknesses have been diagnosed, it can be decided to block access to some functions of the software. The advisor can automatically correct mistakes without advising the user. It can automatically correct mistakes but informing the user. In order to automatically correct mistakes, the advisor must be able to determine what the user wanted to do, even if he or she made some mistakes; the nature of knowledge is not the same used to answer to a given question. For example, think about an individual using a musical writing software; the advisor inferred that the user knowledge level is low. We have a duple time, the unit time is a quarter, so a time is complete when there are 2 quarters or an equivalent. If the user tries to add another quarter, but forgets to put a bar line, a bar line could be automatically put, and, as the user knowledge level being inferred is low, the advisor could inform the user of that correction. This example allows us to point out the need to determine what are the categories of errors and users to which the automatic correction will be applied. If the advisor can both block functions and correct automatically errors, it must have criteria to decide what are the situations for which it has to block access, and those for which it has to automatically correct mistakes. If the advisor can both block functions, automatically correct errors, and present information about mistakes, it must have criteria to decide what are the situations for which it has to block access, those for which it has to automatically correct mistakes, and those for which it has to only present information. Moreover, one has to decide if the criteria will be the same for the different user knowledge levels.

Overall characteristics of the subject of interventions

Taking into account the previous decisions, overall characteristics of the subject of interventions are described. This characterization is made for each resource selected. Does the advisor will do general comments on a procedure without considering the current task carried out by the user? Does the advisor will do comments on the user problem solving process? Does the advisor will only do comments on results of the application of a problem solving process? What is the set of possibilities that will be taken into account?

Overall characteristics of the vocabulary of presentation

The overall characteristics of the vocabulary used to intervene are described; this has to be done for each resource selected. Will the advisor use demonstrations, examples, explanations, index, maps, pictures, procedures, etc? One can formulate his or her own assumptions or refer to learning psychologists' observations. As example, we will refer to Gagné and White (1978). They mention four different memory structures: episodes, images, intellectual skills, and proposition networks. They state that while a particular memory structure can predominate during the learning process, several memory structures are involved during the recall of learned material.

Proposition networks is the memory structure responsible for the representation of sentences or parts of sentences (Kintsch, 1972). These proposition networks are clustered into nodes to represent concepts or events. Recall of those concepts and events consists of the matching of partial patterns, and the construction of sentences using these networks. During the conceptual data modelling process, the user must be able to state the fundamental concepts underlying the process if he or she is to elaborate a model that actually represents the information needed by an organization. The retention and recall of these concepts is closely related to the concept of knowledge stating. According to Gagné and White (1978), knowledge stating is a dependent variable affecting recall; that is the capability to preserve the meaning of a previously memorized text during the elaboration of sentences presenting the topic of this text. Several studies have shown that a person will recall a sentence much better if it is presented within a meaningful context (Coleman, 1965; Downey, and Hakes, 1968; Marks, and Miller, 1964). One can take into account this observation during the elaboration of the explanations. He or she will strive to define concept in regard of a context having significance; for example, any example aiming at illustrating an explanation will be related to the same domain the user is working on.

The memory structure responsible for the identification of concepts and rule application is named intellectual skills (Gagné, and White, 1978). They are the unit components of the afore mentioned learning hierarchy (Gagné, 1968). Contrary to proposition networks which are used for knowledge stating, intellectual skills are used during rule application. As demonstrated by Weigand (1969), the retention of an intellectual skill necessary for rule application requires the retention of subordinate skills. This can lead to consider that the interventions of an advisor must be done in relation with not completely understood intellectual skills. Mechanisms should be provided to identify them. Methods and tools developed in regard of formative evaluation of learners (Scallon, 1988a, 1988b) can be used. Having detected user's weaknesses, remedial actions can then be taken.

There is a relation between proposition networks and intellectual skills. Many studies have shown that the acquisition of an intellectual skill, associated with the verbalization of the new rules discovered, help to improve the retention of the skill and to increase the problem solving capability. Egan and Greeno (1973), Mayer (1974), and Mayer, Stiehl and Greeno (1975) observed that intellectual skill acquisition associated with proposition stating improved the transfer of learned rules to different application fields. Taking these observations into account, an advisor could, during an intervention, present various examples of rule application along with the more traditional statement of the rule.

Images, another type of memory structure, are not necessarily visual. They may be auditory, olfactory, etc., but Gagné and White (1978) state that visual images are generally the most useful. Paivio (1971) observed that images may also come from verbal description of an event or an object. As far as the recall of verbal information is concerned, images have a positive effect (Paivio, 1975). Likewise, recall of relatively long prose passages is improved by the use of images (Anderson, and Kulhavy, 1972; Peeck, 1974; Rasco, Tennyson, and Boutwell, 1975). Dwyer (1970) notes that images seem to have more effect on the short term retention than on the long term retention. As an advisor aims at helping a user while performing a task, the short term memory is the main concern. Therefore, one could decide to include visual images of concrete situations (for example in the field of conceptual data modelling, order forms) within advisor interventions.

The fourth type of memory structure is episodes. They refer to chronological events; they are autobiographical. Episodes corresponds to the following statement: "First I did this..., then I did that..."; it must be distinguished from "First do this..., then do that..." which is an events sequence (Gagné, and White, 1978). One might think that episodes could play an important role during a transaction of help. As example, in regard of the conceptual data modelling process, the advisor reminds a user that in the past he modelled such a situation in such a way. However, the use of episodes during the learning process has not shown any significant improvement of the learning or retention as far as knowledge stating is concerned. As far as intellectual skills are concerned, an equal number of studies are showing positive and no effects. On this basis, one can decide to use or not episodes. That corresponds to answer to the following questions: Will the advisor describe to the user how he or she carried out in order to point what and where is the mistake? Will the advisor describe to the user how he or she carried out in order to point where is the mistake and why it is a mistake?

Knowledge depth

Having established the overall characteristics, they are used to be more specific about the knowledge depth of each type of help transactions: To what level will the advisor help (goal or step)? When will it give details related to a specific step of a problem solving process? When will it decide that it is better to explain all the process (goal)? Will the advisor only describe what to do in regard of a goal or a step? Will the advisor only show, using examples, what to do in regard of a goal or a step? If the advisor uses examples, will these examples be general or related to the specific task currently performed by the user? Will the advisor both describe and show using examples what to do in regard of a goal or a step? Will it be able to answer to questions such as "What would happen if I did this"? or "What is the difference between this two ways of doing this"? Will the advisor consider links with prerequisites when it provides explanations, presents examples, or demonstrates?

Details relative to active resource

When the advisor is an active resource, it must have as components several mechanisms such as tasks recording, tasks analysis, and dialog initiation. What current tasks will be recorded must be determined: Will each key pressed on by the user be recorded? Will the advisor rather associate a set of actions to a taxonomy of errors? How will the advisor do to detect errors? Will it have a record of the way an expert solve a problem allowing it to make a comparison with the user's way? Will the advisor use data issued from historical files? After having made a comparison, will the advisor point out all the differences found or only those related to errors? Will the advisor be allowed to take the initiative of establishing a dialog with the user? If so, what are the decision criteria or cases to be considered? When a user is working, he or she wants to solve a problem. Any intervention criticizing the way of performing a task can be disturbing; the user must not be victim of harassment. Either the user controls the active resource, either a mechanism allowing the advisor to distinguish between desirable and wished interruptions and not desirable and not wished interruptions is installed, or any other mechanism is used. How the explanation will be presented? The advisor should merely suggest a more efficient or more correct way to work, or force the user to perform the complex transfer task in a certain way, the "right" way. What the advisor will do when it is not able to infer any error because the analysis of the recorded data shows a set of tasks not linked together?

Details relative to passive resource

When the advisor is a passive resource, will it only answer to requests? Because the advisor takes control of the dialog after a single request, that corresponds to an unique direction flow of information. Will dialog mechanisms be installed in order to allow the advisor to divide its interventions in such a way that several feedbacks can be received, so the user is in control of it?

Details relative to pedagogical resource

When the advisor is a pedagogical resource, will it present general explanations, or explanations related to the field of the user? For example, an application software being considered, the display of the content of the user guide after a help request in regard of a specific command is an example of general explanation. On the other hand, the explanation of a part of a conceptual database model representing a manufacturing enterprise to a user elaborating the model of such an enterprise is an example of explanation related to the field.

Interface

Several decisions must be taken in regard of the interface. Will the interface be based on natural language? Menu driven? Buttons? Keywords? A mix of these options? Will the user be allowed to switch from a resource to another one? How and when? Who will take the decision? The advisor, the user, both? If the advisor can decide, what are its decision criteria?

Table 2.2 summarizes the main decisions to be taken during the first step of the development life cycle.

Table 2.2

Summary of the first step

1.	Aim of the advisor.
2.	Target users of the advisor.
3.	Type of resource.
4.	Main characteristics of the course of interventions.
5.	Way used by the advisor to adapt itself to the user.
6.	Overall characteristics of the subject of interventions.
7.	Overall characteristics of the vocabulary of presentation.
8.	Knowledge depth.
9.	Details relative to active resource.
10.	Details relative to passive resource.
11.	Details relative to pedagogical resource.
12.	Interface.

The two following chapters present two case studies. The content of the files related to the first step of the development life cycle is presented. Note that a brief description of the domain precedes the presentation of the results of the first step. The content of the files is presented according to the order used at table 2.2. As it is possible that one or several topics included in the table be not relevant, one or many items of table 2.2 may not appeared in the case studies.

CHAPTER THREE

CASE STUDY 1

CONCEPTUAL DATABASE MODELLING ADVISOR SYSTEM: FILES RELATED TO THE DESCRIPTION OF THE STRATEGY OF INTERVENTION

Conceptual database modelling domain

Because it is a complex transfer task, to teach and to learn conceptual database modelling is very difficult. It is not enough to follow the numerous principles and rules to elaborate the proper solution, i.e., the conceptual database model that does represent the various access to data of the database required by an organization. When elaborating a conceptual database model, perception of details is as important as the use of principles and rules. To exercise learners in perceiving details, cases are used.

A conceptual database model is like a picture of an organization's memory. A diagram represents the information the organization needs in order to operate; it also shows how those pieces of information are linked. Conceptual database models are concerned with data meanings.

The conceptual database modelling comes from the need to manage information circulating within and without organizations. To meet the needs managers have for information, a conceptual database model is elaborated. Such a model separates information from the processing it undergoes. The more accurately the elements of information reflect the reality of the organization, being what are the different pieces of information it needs to record, the more useful the model and the database that will be developed later are.

A conceptual database model aims at representing all the primary informations within the organization as well as all the processed informations that the organization must record. A primary information is a raw information that has never been processed. Because a total is derived by processing primary informations, it is an example of processed informations. If an organization needs to record a total for any reason (such as legal reason), it should be represented in the conceptual database model. The analyst has to question the people responsible for the organization to decide how this total should be represented in the conceptual database model.

A conceptual database model is elaborated by identifying all pieces of information that circulate. A diagram shows the identifiable items, named entities, and the links, named relationships, joining these identifiable items (figure 3.1).

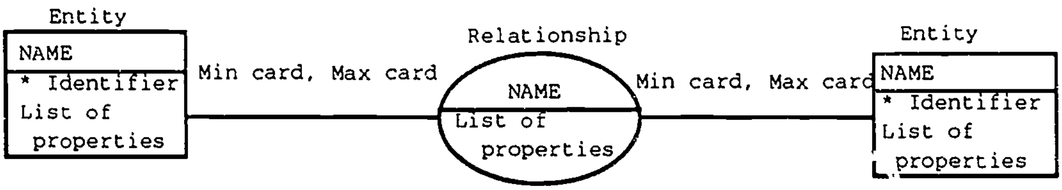


Figure 3.1 Entities with a relationship between them (MERISE formalism).

These identifiable items and the links between them are described by properties. To clarify the meaning of the relationships, cardinalities are used. Another element of a conceptual database model is the constraints. They influence how the conceptual database model is used and how understandable it is; constraints let represent the reality more accurately.

Although each conceptual database modelling technique proposes its own modelling rules, how an analyst decides to represent a specific reality is heavily influenced by the way he or she perceives the reality of the organization. The concept of marriage will be used as example (Tardieu, Nanci, and Pascot, 1984). This concept can be represented in different ways; the correct one is found by perceiving the needs of the organization for which the conceptual database model is elaborated.

The concept of marriage can be represented by a single entity MARRIAGE, described by the properties MARRIAGE REGISTRATION NUMBER, MARRIAGE DATE, WIFE'S SOCIAL SECURITY NUMBER, and HUSBAND'S SOCIAL SECURITY NUMBER (figure 3.2).

MARRIAGE
* Marriage reg. nb.
Marriage date
Wife's soc. soc. nb.
Husband's soc. soc. nb

Figure 3.2 Marriage concept represented by the single entity MARRIAGE.

When the concept of marriage is represented in this way (figure 3.2), each wedding is a new occurrence of the entity MARRIAGE; only one marriage date is possible. That means the organization solely needs to keep a record of each ceremony. If the organization needs to know with whom a person is married, using a single entity to represent the marriage concept is inadequate. Therefore, the concept should be represented as illustrated at figure 3.3.

When this representation is used (figure 3.3), the organization can easily access the information it needs, being with whom a person is married even when a person gets married more than once. But if a person marries again an ex-spouse, this information will overwrite their previous marriage date because the identifier of the relationship is the same in both cases. Hence, the newly-formed couple is undistinguable from the old one. If the organization needs to keep track of this special case, another representation would be required (not illustrated here).

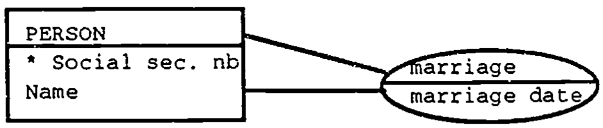


Figure 3.3 Marriage concept represented by one entity and one relationship.

Whether to represent the reality of the organization with entities or relationships is determined by the needs of the decision-makers, i.e., what information they need to have access to. The above example illustrates one of the difficulties related to the teaching and learning of conceptual database modelling.

A conceptual database modelling course has two parts: a theoretical part in which concepts, principles, and rules, are presented, and a practical part in which students have to apply the theoretical concepts, principles, and rules, when resolving cases describing different organizations and their data. At the end of their curriculum, students have a training period within an organization. CODAMA was developed to help them at the end of the course, and in the real world.

Aim of the advisor

CODAMA aims at helping a user of a graphic software to elaborate a conceptual database model. Transfer of knowledge is the main concern: transfer related to the general process of conceptual database modelling and to the elaboration of such a model for a particular sector of activities (manufacturing, government, hospital, etc.).

Target users of the advisor

There are several categories of users: 1- Users having weak knowledge in regard of concepts, principles and rules, and no practical experience; 2- Users having satisfactory knowledge in regard of concepts, principles, and rules, but having had only few opportunities to apply them, they experience problems faced with the transfer of knowledge; 3- Users having satisfactory knowledge in regard of concepts, principles, and rules, having had in the past opportunities to apply them when they elaborated the conceptual database model for organizations acting in a specific sector of activities, but having no experience related to the sector of activities for which they currently elaborate such a model.

Type of help transactions

CODAMA provides three types of help transactions: active, passive, and pedagogical.

Main characteristics of the course of interventions

Active resource

When CODAMA is an active resource, it merely checks if the rules of the formalism are properly applied.

Passive resource

When CODAMA is a passive resource, it answers to requests in reagrd of the conceptual database model being currently elaborated.

Pedagogical resource

When CODAMA is a pedagogical resource, the pieces of knowledge allowing to set a particular topic in the frame of prerequisites are presented. In its presentation, CODAMA takes into account either a sequency of prerequisites, or the weaknesses discovered when the user transferred the knowledge in elaborating a part of, or a complete conceptual database model.

How the advisor adapts itself to the user

Active resource

When CODAMA is an active resource, it considers the nature of errors. They are associated to an objective being part of a hierarchy.

Passive resource

When CODAMA is a passive resource, the domain is considered as being a set of prerequisites. Individualization is function of the nature of questions asked, the frequency of requests for help for a same topic, the user feedback as an explanation goes along, and the frequency of errors made being calculated from data recorded by the active resource. Dependent on the values taken by those variables, some interventions may not be available for a certain period of time.

Pedagogical resource

When CODAMA is a pedagogical resource, its capacity to adapt itself to the user is function of not mastered prerequisites. It uses the following data to infer what are those prerequisites: frequency of the requests for help for a same topic (from passive resource), user feedback received as an explanation goes along (from passive resource), nature of questions asked (from passive resource), nature of errors (from active resource), and the frequency of each error (from active resource).

Overall characteristics of the subject of interventions

Active resource

As an active resource, CODAMA only takes into account the result, i.e., the conceptual database model elaborated. It merely does comments on errors, so it does not take into account the method used to solve the problem.

Passive resource

As a passive resource, CODAMA merely does comments on the different elements of the conceptual database model elaborated by the user; consequently, it does not take into account the method used.

Pedagogical resource

As a pedagogical resource, CODAMA is able to explain both the whole process and a particular step of the elaboration of a conceptual database model. Its explanation does not take into account the current model being elaborated by the user, neither the method used.

Overall characteristics of the vocabulary of presentation

Active resource

When CODAMA is an active resource, its explanations refer to a meaningful context. It uses the different elements of the conceptual database model currently elaborated by the user (i.e., properties (name), entities (name, identifier and list of properties), relationships (name, identifier, list of properties and number of legs), cardinalities (meaning), and constraints (meaning)), to state its comments; comments point out the prerequisites.

Passive resource

When CODAMA is a passive resource and does an intervention in regard of a specific element of the conceptual database model elaborated by the user, it uses the information related to this element (i.e., properties (name), entities (name, identifier and list of properties), relationships (name, identifier, list of properties and number of legs), cardinalities (meaning), and constraints (meaning)), to formulate questions and explanations related. They both refer to a meaningful context. References to prerequisites are outlined in each explanation. CODAMA also receives a feedback from the user.

Pedagogical resource

When CODAMA, as a pedagogical resource, displays an explanation in regard of a concept, principle, or rule, one or several examples related to the same kind of organization being modelized (hospital, manufacturing, service, etc.) can also be displayed, if it is useful to do so. When examples are displayed, that corresponds to a meaningful context. References to prerequisites are made clear both in explanations and examples. If it is possible to do so, forms used in the organization will be displayed in order to provide images of concrete situations.

Knowledge depth

Active resource

In regard of the knowledge depth of the active resource, CODAMA considers the goal; it does comments on the whole conceptual database model. When errors are related to one element of the conceptual database model (for example, constraint), CODAMA will solely consider a step, i.e., this specific element.

Passive resource

In regard of the knowledge depth of the passive resource, CODAMA considers only steps. When it does comments, it considers only one element of the conceptual database model at a time.

Pedagogical resource

The knowledge depth of the pedagogical resource refers to two levels: goal and steps. When CODAMA produces an intervention related to the whole process of elaborating a conceptual database model, it considers goal. When CODAMA produces an intervention in regard of a specific element, it considers step.

Details relative to active resource

The user decides when CODAMA becomes an active resource. It only does comments on errors. As each comment relates to the current conceptual database model, there is no specific task to record. The errors detection is made by checking the proper use of the rules of modelling, and by a comparison with a list of elements issued from historical files. In that list, there are common words used as identifiers, the kind of organization being considered, (for example, social security number), and statistics establishing the frequency of use of each individual cardinality.

Details relative to passive resource

When CODAMA is a passive resource, it proposes a list of questions, presents an explanation in regard of each of these questions, and waits user feedback. CODAMA takes into account the nature of feedback, the number of requests made in regard of a same topic, and the time elapsed, to determine the next list of questions to be displayed.

Interface

Active resource

The interface of the active resource is mainly based on natural language and keywords. Natural language refers to pre-stored sentences in which variables are introduced; the content of these variables refers to names of elements being used in the conceptual database model elaborated by the user. Mechanisms are installed to take into account gender and number. To allow the user to have a direct access to the pedagogical resource, some words are outlined. These words refer to prerequisites. The user only has to select one of those words to switch to the pedagocical resource, allowing him or her to have the explanation related. This access to the pedagogical resource is exclusively controlled by the user.

Passive resource

The interface of the passive resource is based on menus, natural language, buttons, and keywords. There are several levels of menus, and some menus use natural language. At the first level, the user selects the topic (for example, cardinality). The topic being selected, CODAMA, before displaying the next menu, asks the user to point out the element(s) for which he or she wants an explanation. It uses the name of these elements (for example, an entity named student, another entity named course, a relationship named enrolls, etc.) to replace the variables in the pre-stored questions. A question having been selected in the menu, the explanation associated is a pre-stored sentence having variables; the names of elements replace the variables. In both cases, mechanisms related to gender and number are installed. When explanations are presented, buttons are used to obtain user feedback. The user can also switch to the pedagogical resource; he or she merely selects a keyword outlined in an explanation. This access to the pedagogical resource is exclusively controlled by the user.

Pedagogical resource

The interface of the pedagogical resource is based on menus, and keywords. Several types of interventions are possible. First, there are explanations displayed according to a precise request; this request can be made by selecting keywords in the explanations given by one or another resource, or by selecting a question in the pedagogical resource menu. There are several levels of menus. At the first level, the user selects the topic (for example, entity). The topic being selected, CODAMA displays the next menu that proposes the list of questions related. The user selects whatever numbers of questions needed. Note that when the user comes from active or passive resource, i.e., when he or she selected a keyword, this user has a direct access to the explanation corresponding to the keyword. He or she does not have to use menus. In the same way, the user can use menus to have a first explanation and keywords for the next ones.

The user can also ask CODAMA to plan a presentation of the content taking into account his or her performance (or his or her lack of performance); it is a made-to-order training. CODAMA will use the knowledge of the user it has recorded to individualize the content. If the user did not use any resource, CODAMA will hypothesize that the user wants a general presentation of the process of elaborating a conceptual database model (Advance organizer as defined by Ausubel, 1960). First, CODAMA displays a complete conceptual database model. This model being displayed, it simulates, according to the hierarchy of prerequisites, how this model was elaborated. According to principles of conception of advance organizers (Guéladé, 1987), the length of these explanations will be the quarter of those displayed when the user selects a question in a menu. If the passive resource has been used, but not the active resource, (i.e., the user did not make errors), CODAMA will use data recorded by the passive resource and the hierarchy to produce the advance organizer. If there are data recorded by the active resource, that means that the user made errors. These data will be used to produce a remedial teaching. In the same way, a remedial teaching will be proposed to the user who did use both active and passive resources. A larger number of data will be taken into account when producing the remedial teaching. Note that the user may use keywords to modify the order of presentation.

Note

When a user, in the active or passive resource, selects a keyword in order to have a precise explanation, he or she is always under the control of this resource. The user can select as many keywords as necessary, but soon or later, this person will have to return to the resource he or she comes from.

CHAPTER FOUR

CASE STUDY 2

MUSICAL COMPOSITION ADVISOR SYSTEM: FILES RELATED TO THE DESCRIPTION OF THE STRATEGY OF INTERVENTION

Context

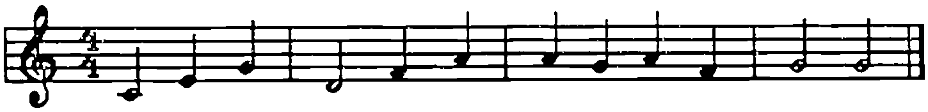
In 1981, the Government of Québec (Ministère de l'Éducation, 1981a, 1981b, 1983) adopted new policies that affected the teaching of music at the primary and secondary levels. The Québec Ministry of Education describes the various components of the music teaching for the two academic cycles; general objectives, end objectives, and intermediate objectives with associated knowledge are stated. The music program is made up of six modules: creation, execution, graphics, musical language, musical literature, and sound environment. The current file relates more specifically to the creation module of the third secondary level; it aims at allowing the student to transfer his or her knowledge of the musical language. Creation is considered in accordance with Burns (1988):"... the creative process not only synthetizes previous learning but also elevates the mind into a higher stage of reasoning - the problem solving stage" (p.62), and Moore (1990): "The tasks ... were conceived as "musical problem solving" within a context of implicit as well as explicit musical parameters (an incomplete melody)" (p.28).

Aim of the advisor

In the music program proposed by the Québec Ministry of Education (Ministère de l'Education, 1981b), the general objective is formulated as follows: To create his or her own music using the sound material (p.48). The terminal objective is: To compose according to some parameters (Terminal objective 9.2.1). The intermediate objective is: To invent short melodic themes, and develop them using processes such as augmentation, diminution, repetition, variation, accentuation changing, retrogradation, inserting or subtraction of notes or silences (Ministère de l'Éducation, 1981b, p.48). As associated content, we find: tonality, cadence, form, range of the melody, question-answer, starting on tonic, third or fifth degree, starting or not on the first beat (Ministère de l'Éducation, 1981). MUSIC aims at helping a student using a musical writing software while doing his or her creation activities. Transfer of knowledge is the main concern. Statements of exercises related more specifically to the question-answer process are inspired by Musicontact 3 (Fournier, Milot, Richard, Béchard, and De Melo, 1986):

1- Make up an answer to the following question:

Parameters:
- The range must not exceed an octave
- The melody must end on the tonic



2- Starting with the key signature, time signature, and first note indicated at the beginning of the stave, make up a melody having eight bars in accordance with the question-answer process. The range of the melody must not exceed a fifth.

The statement of exercises related to augmentation, diminution and repetition processes is also inspired by Musicontact 3 (Fournier, Milot, Richard, Béchard, and De Melo, 1986)[1]:

3- Here is a melody; transform it using augmentation, diminution, repetition processes.

Target users of the advisor

The target user of MUSIC is an individual to who the musical language elements were taught. The knowledge taught corresponds to the third secondary level described in the secondary music program (Ministère de l'Éducation, 1981b, 1983).

Type of help transactions

MUSIC is a passive resource, i.e., it waits a request from the student to intervene.

Main characteristics of the course of interventions

While the student composes and in accordance with his or her request, MUSIC answers questions in regard of musical language elements, does comments on the composition, creates its own examples, and does comments on them.

---

[1] In the real files, each process of composition is fully detailed. Note that in this book, we will merely present content of files in regard of exercise 1. The whole analysis can be found in Boulet and Dufour (1991), and in Boulet, Dufour and Lavoie (1991a, 1991b). This note is valuable for all the steps.

How the advisor adapts itself to the student

Individualization is ensured by considering student's requests and composition. Comments related to student's composition, the process of composing its own example and doing comments on it, are done in regard of rules issued from tonal system, and those stemming from the musical composition process currently used.

Overall characteristics of the subject of interventions

MUSIC will merely do comment on results, i.e., the composition made up.

Overall characteristics of the vocabulary of presentation

Taking into account the main characteristics described before, interventions have been categorized in the following way: 1- explanations requested by a student and related to musical language elements, 2- comments on examples MUSIC composed, 3- comments on the student's composition. Explanations will present definition of concepts, or statement of principles or rules; in both cases, an example will be presented, when suitable. Melodies created by MUSIC will relate to the musical composition process currently used: for example, the composition of a G Major melody of eight bars, using the question-answer process, time signature being $\frac{4}{4}$, range of the melody of an octave. Comments on the student's composition will be done in regard of the use of musical language elements.

Knowledge depth

Solely the goal is considered, i.e., to compose a melody. MUSIC does a global comment on the whole composition (its own or the student's one); it does not take into account the method used. It always refers to the composition being commented. It does a comparison of the use of the musical language elements with rules of tonal system. Links with prerequisite capabilities are made. In regard of explanations presented at the student's request, examples are general; they are merely used to illustrate the definition of a concept, or the application of a principle or a rule.

Details relative to passive ressource

MUSIC merely answers to the student's request. Consequently, explanations or comments do not request feedback.

Interface

MUSIC's interface is a mixed one: menus, keywords and buttons. Natural language is also used when MUSIC does comments on a composition. Comments are pre-stored sentences having variables. Content of those variables is adjusted considering the characteristics of the composition. Mechanisms related to gender and number are installed. Buttons and menus allow the selection of a particular option. Keywords are used to select a given explanation.

Precision

MUSIC will be used within a scholastic context. With the passing weeks, activities of composition will be more and more complex, taking into account the new knowledge learned by students.

CHAPTER FIVE

STEP 2. DESCRIBING THE KNOWLEDGE ACQUISITION STRATEGY

Boundaries of the domain

At this step, the content is the main concern. Means that will be used to delimit the domain are described. That corresponds to analyze and characterize the content of the transfer task. It is possible to set out his or her own assumptions, or to refer to one or many instructional or learning theories. Because members of the project team will have to be trained, when one decides to set out his or her own assumptions, all theoretical elements related must be detailed and well-documented. The use of existing methods, principles, theories, tools, etc., allows time saving and makes easier the recruitment and training of each member of the team. Whatever option is selected, reasons behind the selection of an approach, method, technique, theory, tool, etc., must be carefully detailed.

Sources of knowledge: difficulties

Means and sources that should be used to identify difficulties an individual may have, when he or she performs a complex transfer task X, while using an application software Y, are selected and described.

The use of a standardized test can be proposed in order to identify the nature of the not mastered knowledge. If such a test is available, values associated to its reliability and validity (such as level of confidence) can be used to estimate what proportion of target users will not be actually helped by the advisor (because some of their difficulties are not part of the test). This value can then be used to estimate the cost of errors that users could make despite the investment in the technology. If it is decided to use a standardized test, but such a test is not available, it has to be developed. The test being elaborated, steps of the process allowing the estimation of its reliability and validity must be done. Again, the utility of the investment will then be estimated.

It can also be decided to analyze how a representative sample of target users performs the transfer task. Means that will be used must be detailed. Will the performance be observed, or shooted? We can ask these users to state their questions out load; it will then be possible to make a list of their questions. Because irrelevant variables can affect the environment, the collection of data must be rigorously surrounded. Statistics will be produced in regard of the proportion of users that the advisor will fail to help (because their questions were not collected), using values such as the proportion of users that may not be represented in the sample; it will allow the estimation of cost of the advisor's failure, utility of the advisor, etc.

In order to collect data allowing the identification of user's help requests, it can be decided to set a simulation of the functioning of the advisor. An analysis of protocol will then be done. Data collected can be categorized. (An example of simulation followed by an analysis of protocol to categorize users' questions can be found in Parent (1990)). Means to ensure that the sample actually covers the variety of future users must be taken. Means to ensure that the (subjective) process of linking a request for help to a category will be objectively done have to be planned. To avoid a too large lost of information (simplification of the reality), number of categories used to classify the requests for help must be appropriate. Here again, statistics can be established: proportion of users that are not represented in the sample, reliability and validity of the case study with error associated, etc. The various values will then be used to estimate the proportion of future users that will not be helped by the advisor (because their questions were not collected so, the advisor is not able to answer); the linked risk, i.e., they make mistakes, will be estimated. Cost related (such as for the manufacturing sector, a lot of imperfect products, and for the academic sector, students that cannot properly write a text (cost for the whole society)) will be estimated too.

This list of possibilities does not intend to be complete; a mix of several techniques is also possible, especially if the margin of error must be minimized. Two examples of context where it is possible that this margin may be set at zero are: an environment where any human error may threaten the health and security of persons, or an organization being involved in a total quality program.

Sources of knowledge: interventions

Means and sources that will be used to write content of interventions the advisor does, when an individual performing a complex transfer task X, while using an application software Y, makes a request for help or mistake, are selected and described.

There are several possibilities. Will specialized manuals be used? Will course syllabi be used? If so, will those syllabi come from academic sector? Will those syllabi come from professional sector? Will content of courses given in the academic sector be analyzed? Will content of courses given in the professional sector be analyzed? In both cases, how it will be done. Will these courses be observed, or shooted? Will experts performing the task and expressing out load how they perform the complex transfer task be observed, or shooted? Will a simulation of the advisor be setted in order to be able to do an analysis of protocol allowing the identification of experts' answers facing a user's request? Will the data be categorized? Cautions mentioned before in regard of the users' requests for help identification and categorization are relevant. Those mentioned in regard of calculating values to be used to estimate the margin of error of the advisor (i.e., what proportion of users the advisor will fail to help) are also relevant. Decisions on how data collected in regard of explanations given by experts will be processed must be taken. Analysis of these data must allow the identification of the really useful explanations, i.e., those that actually helped the user to perform the complex transfer task. Just remind that the development life cycle proposed in this book is user oriented. Therefore, measures aiming at determining the usefulness of each explanation must be taken. Values issued from statistics can then be used to estimate the proportion of users that will not be helped even if the advisor is able to answer their precise question. In the same way, the cost related can be estimated.

To determine what will be the content of interventions, it can be decided to refer to a certain form recommended by researchers. Those researchers recommend it after having done several rigorously controlled experiments involving several samples; these experiments were done to identify how to facilitate the mastery of a particular capacity described by the mean of an action verb. As example, we mention instructional tactics proposed by Leshin, Pollock and Reigeluth (1992) in regard of each type of capacity they define. The Gagné and Briggs' (1979) principles of instructional design being stated after several years of experiments are another example. The use of such principles or tactics can contribute to minimize the margin of error. Moreover, the values necessary to estimate the utility and the cost of failure can easily be found in the literature related.

To summarize, at this step, sources and models that will be used to delimit the domain are identified. Then, means to use to collect knowledge (difficulties, and interventions) are selected. Means to assess quality of this knowledge are proposed: characteristics of the sample, means of control used when categorizing, usefulness of explanations, etc. Means to evaluate margin of error of the advisor are also proposed. In regard of further researches, as the form of explanations will be carefully detailed and documented, many of them might be compared, equivalent samples of users being exposed, in order to find which is or are the most effective. Values to be used to estimate the risk might also be calculated.

The knowledge acquisition method

Having selected sources of knowledge and means, steps of the corresponding knowledge acquisition method are then detailed. Note that the method is not a generic one.

Table 5.1 summarizes the main decisions pertaining to this second step.

Table 5.1

Summary of the second step

1.	To select and describe means to delimit the boundaries of the domain.
2.	To select and describe means and sources of knowledge in regard of the difficulties.
2.1	To select and describe means to insure quality of the data collected (difficulties).
2.2	To select and describe means to evaluate utility and risk (difficulties).
3.	To select and describe means and sources of knowledge in regard of the interventions.
3.1	To select and describe means to insure quality of the data collected (interventions).
3.2	To select and describe means to evaluate utility and risk (interventions).
4.	To set out the knowledge acquisition method.

Content of the files

The content of the files issued from this step begins by presenting a summary of the main aspects found in step one files. Means that will be used to fill the requirements of these main aspects are detailed; reasons behind the decision to recommend that a task be performed in a particular way must also be stated. These details might be used to identify what abilities members of the project responsible for the knowledge acquisition or for the elaboration of the content should have. If necessary, a training plan will be foreseen. It will also ensure that the work is being carried out in accordance with what is expected, avoiding thus delivery time being brought about by works started again. For the same reason, steps of the method proposed for the knowledge acquisition and the elaboration of the content are presented.

General remarks

At all times, one must keep in mind that the idea is to be "inspired by"; the idea is not to "force anybody". Ideally, the head of project must develop a group culture in accordance with this idea. Approaches, means, methods, techniques, tools, etc., recommended must not be blindly used. They are not limits, but guides, helps. Instead of "reinventing the wheel", ideas of persons who had a deep thinking about an aspect are applied; their ideas are in fact adapted. To do so, members of the project team have to do a transfer of knowledge. That is what the training plan must aim at. We lay great stress upon the fact that recommendations must be considered as helps, and not as obligations to conform to a technique; adaptations are possible. In fact, recommendations may be seen as an attitude; they are sources of inspiration. Flexibility, adaptability are attributes of the main idea.

The performance of the tasks pertaining to this step can generate (and will generate) modifications of the content of files issued from the previous step.

CHAPTER SIX

CASE STUDY 1

CONCEPTUAL DATABASE MODELLING ADVISOR SYSTEM: FILES RELATED TO THE DESCRIPTION OF THE KNOWLEDGE ACQUISITION STRATEGY

Main aspects of files issued from step one

- Division into hierarchies of prerequisites
- Interventions in regard of prerequisites
- User's weaknesses diagnosis
- Production of remedial help
- Use of examples

Division into hierarchies of prerequisites

According to files issued from the previous step, prerequisites are the foundation of the knowledge CODAMA records and uses. A hierarchical division of the domain has to be done.

Description of means recommended to divide the domain

To divide the domain into prerequisite capabilities, the use of Gagné and Briggs' principles of instructional design is recommended (1979; Brien, 1980).

According to those principles, the goal is formulated in terms of behavior and content. The formulation must include all the capacities that will be mastered at the end of the training (Brien, 1980). According to the purpose of CODAMA stated at the previous step, we can make out the following goal: the user must be able to elaborate a conceptual database model.

Afterwards, according to Gagné and Briggs (1979), objectives of the different parts are defined, the goal being considered. There are several categories of objectives: unit objectives, end objectives, and intermediate objectives.

Unit objectives are statements that take into account groups of other capabilities that a person should master to be able to reach the goal of the course (Brien, 1980). The unit objectives are identified in figure 6.1 by the prefix U.O. followed by a number; this number represents the rank of each objective within the hierarchy. Each unit objective is separated into end objectives (t at figure 6.1). They represent capabilities which suppose the master of other capabilities, called intermediate objectives (i at figure 6.1). To identify them, it is advisable to refer to Ausubel's works (1968) related to links between propositions and to Gagné's works related to the analysis of intellectual skills (1962, 1972, 1977).

These three types of objectives having been stated, each of them is linked to a category proposed in one or many taxonomy(ies). Examples are: idea, proposition, interrelated set of propositions, discrimination, concept, rule and higher-order rule (problem solving)[1]. An illustration of links between unit, end, and intermediate objectives, is presented at figure 6.1.

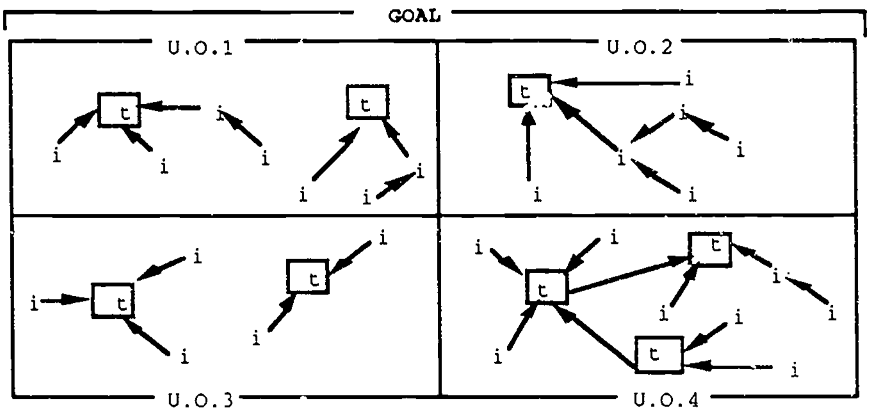


Figure 6.1 Links between objectives (Adapted from Brien, 1980).

Summarizing the information presented at figure 6.1, the "t" and "i" represent the various capabilities any person must acquire to be able to reach the goal. The unit objective one capability must be acquired before a person can perform the task corresponding to unit objective two; the unit objective two capability must be acquired before a person can perform the task corresponding to unit objective three, and so on. Within each unit, many situations can occur. Two end objectives may not be prerequisite capabilities; end objectives of unit objectives 1 and 3 are examples (figure 6.1). One or several intermediate objectives may have as prerequisites one or several other intermediate objectives; it occurs for unit objective 2 (figure 6.1). One end objective can also have as prerequisites one or several other end objectives; this type of link is illustrated at figure 6.1 within unit objective 4.

---

[1] In the real file, the taxonomy refers to Ausubel's (1968) and Gagné's (1977) works. As it was presented in chapter two, this part of the file is not presented.

Interventions in regard of prerequisites

The hierarchy that corresponds to the elaboration of a conceptual database model capacity being settled, it is possible to write the statements of questions, rules, and explanations, in regard of each element of the hierarchy, in other words in accordance with each intermediate objective. As mentioned in the files of the previous step, questions displayed by the passive and pedagogical resources and explanations associated, comments on errors done by the active resource, or keywords used in the three resources, are the raw material of the user/CODAMA transactions. That involves that links between the knowledge divided into intermediate objectives and comments, explanations, keywords, and questions, be created. To create them, the first version of the rules to be verified, of the questions, and of the explanations related, will be stated according to each intermediate objective. Keywords are used to provide an easy access to a specific explanation without using the hierarchical sequency. We recommend to associate a keyword to each explanation. To represent those links, a network will then be elaborated. When the content of explanations is stated, authors will have to make sure to use everywhere the same keyword when referring to a same explanation.

User's weaknesses diagnosis and production of remedial help

Analyzing the content of the files issued from step one, we found that two concepts on which instructional researches have been done are relevant: formative evaluation and remedial teaching. In regard of the passive resource, questions asked by and feedback received from the user help to pace each user's progress; the content of menus proposed by the passive resource is adapted in function of these data. In the same way, the pedagogical resource being considered, a made-to-order training is available to the user: the content of this special training is elaborated in regard of the analysis of the data provided by the active resource, the passive resource, or both. The idea of pacing involves the use of appropriate remedial means to help users get back on track when they run into trouble. The terms "pacing" and "remedial" are part of the formative evaluation of learners concept (Bloom, Madaus, and Hastings, 1981).

Formative evaluation of learners involves the use of instruments designed for the specific purpose of continuous intervention in the progress of each student (Scallon, 1988b). What a student has learned at a given moment is compared with what the student should have learned by that time (Barbier, 1985). Bloom, Hastings and Madaus (1971), Fontaine (1979), and Scallon (1988b) mention that the main purpose of formative evaluation is to help a student making progress. This purpose is in accordance with the advisor one, being: To help the user of an application software when he or she performs a complex task that involves a transfer of knowledge. The function of formative evaluation is to identify as soon as possible difficulties and to inform the student of sources and causes of his or her difficulties (Airasian, and Madaus, 1972; Bloom, Hastings, and Madaus, 1971; Noizet, and Caverni, 1978). This function is similar to the advisor one, being: To infer what difficulties a user faces and to use different strategies to allow him or her to get back on the right track.

To be able to apply principles related to formative evaluation of learners, end and intermediate objectives must have been be identified (Bloom, Madaus, and Hastings, 1981; Scallon 1988b). The previous recommendation of using Gagné and Briggs' principles of design involves that such objectives are formulated; therefore, this requirement is fulfilled.

Remedial teaching begins with an exact diagnosis of why a student is facing learning problems. Ways to overcome these problems are then proposed (Block, 1977; Block, and Anderson, 1975; Bloom, 1978a, 1978b; Bonboir, 1970; Cheek, and Cheek, 1980; Provencher, 1985). Remedial teaching is based on the diagnosis supplied by formative evaluation. In regard of CODAMA, files issued from step one reveal that the pedagogical resource has to be able to propose a made-to-order training; its content will be planned in accordance with the diagnostic information collected by the active and passive resources. This kind of intervention links totally with the remedial teaching concept.

Having established that CODAMA would have to be able to use formative evaluation tools to diagnose users' transfer of knowledge problems, and then provide appropriate remedial help, the next step was to take a look at the tools.

Description of means proposed to diagnose weaknesses and produce remedial help

Because the kind of capability to evaluate determines which formative evaluation tool is relevant, and taking into account that an intelligent advisor is an adaptive system aiming at helping the user of an application software while performing a transfer task, elaborating a conceptual database model is an intellectual skill (as defined by Gagné, 1972); more specifically, it is an higher-order rule (problem solving).

At this point, the implication the recommendation to get inspired by Ausubel's (1968) and Gagné's (1977) works has in regard of the acquisition and representation of the knowledge can be drawn. Rather than modelling the knowledge by solely determining what are the links between various concepts without regard of the characteristics of a task or how they are used when a person performs the task, it will be modelled in the following way: first, the knowledge will be defined in terms of action verbs corresponding to a capacity; then, it will be organized in several linked tree structure taking into account what an individual knows when he or she is able to perform the task characterized by the action verb. Thus, we accept that the most important condition for ascertaining whether an individual has learned something is whether or not the individual has learned the prerequisites (Gagné, 1962, 1968; Gagné, Mayor, Garstens, and Paradise, 1962; Gagné, and Brown, 1961). We concur with the above authors when they state that each person approaches a task with a different set of previously acquired capabilities, and in order to be effective, any teaching program must take into account what this person can and cannot do. To be able to take it into account, formative evaluation tools will be used. We also agree with the above authors when they state that it is very important to teach the prerequisites to an intellectual skill that the person has not yet mastered, and to enable the person to review these prerequisites at any time. To do so, CODAMA must have the ability to link user's errors (active resource), and questions the user asks or is asked (passive resource), to an objective being part of a hierarchy. We recommend that formative evaluation tools based on the criterion approach and the domain referenced testing be used to elaborate a picture of the user profile in regard of his or her difficulties, allowing CODAMA to propose either a relevant remedial action, or a relevant form of help; more specifically, it is proposed to get inspired by the multiple facet scheme (Guttman, and Schlesinger, 1967a, 1967b; Scallon, 1988a), and the hierarchical test (Scallon, 1988a).

The main reason supporting the recommendation to use the criterion approach can be found in Radocy and Boyle (1987): "Essentially the decision to use a criterion or norm-referenced approach for assessing student achievement depends on the particular function that the test is intended to serve. If the concern is to assess the individual student and obtain feedback about his or her achievement of a relatively specific content, the criterion-referenced approach is the more appropriate" (p.33). The criterion approach is characterized by Glaser (1963) as follows: "Measures which assess student achievement in terms of a criterion standard thus provide information as to the degree of competence attained by a particular student which is independant of reference to the performance of others" (pp.519-520). As far as CODAMA is concerned, these two observations are linked to abilities that it must have.

The criterion approach brings to domain referenced testing and hierarchical test. Domain referenced testing is a qualitative approach aiming at describing in a very precise way the content of a test (Hively, 1974). It is a tool in continuation of the criterion approach (Scallon, 1988b).

Domain referenced testing integrates a set of methods; multiple facet scheme (Guttman, and Schlesinger, 1967a, 1967b) is one of those. According to Scallon (1988a), the determination of performance profiles based on the characteristics of a task is made in a more rigorous way when the specification of the domain is elaborated according to the facet theory.

The multiple facet scheme provides a precise way of evaluating how close an individual is to attaining a learning objective of the intellectual skill type. It is designed to help to decide whether a student should be allowed to go on other objectives, or be given remedial instruction for the specific problems identified. This model uses facets to define a group of items and derives from them a specification table. During the knowledge acquisition process, we recommend that, when possible, such a table be developed for intermediate objectives. The same table will later be used by CODAMA to elaborate the user model.

The elaboration of a multiple facet scheme and of an analysis table to compile the results process follow three distinct steps: subject analysis, writing the test, and elaboration of the correction tool (Boulet, Labbé, Laprise, Lemay, and Lavoie, 1987; Boulet, Labbé, Lavoie, and Lemay, 1989).

Subject analysis

1- Identify the domain being evaluated: Minor musical scales will be used as example.

2- Define the scope of the domain being evaluated: That corresponds to provide an answer to the following question: What are the difficulties a student could face if he or she has to write a minor scale. This question must be repeated until all dimensions of the domain be identified. Two major difficulties can be identified in the domain used as example (the minor scales): the student should be able to write a melodic minor scale and a harmonic minor scale (figure 6.2).

HARMONIC	MELODIC

Figure 6.2 Identification of the first two dimensions.

This division being considered, it can be observed whether a student is facing difficulties when he or she writes harmonic minor scales, melodic minor scales, or both.

By asking more questions, it can be determined that the student can not only face difficulties while writing harmonic or melodic minor scales but also while writing ascending or descending minor scales. Two dimensions are added: ascending and descending as illustrated at figure 6.3.

HARMONIC		MELODIC	
ascending	descending	ascending	descending

Figure 6.3 Adding dimensions ascending and descending.

Another division of the domain can be identified in order to be able to observe whether a student has difficulties with minor scales possessing flats, those possessing sharps, or both. This division is illustrated at figure 6.4.

	HARMONIC		MELODIC	
	ascending	descending	ascending	descending
Sharp				
Flat				

Figure 6.4 Third division of the domain.

Other divisions like the location of tone and semitone could be added. Note that for the purpose of the example, the number of divisions being taken into account is reduced.

3- Identification of the facets of the domain: The dimensions of the domain of minor scales having been determined, one has to identify to which facets each dimension is linked. Using the same example, the dimensions "harmonic" and "melodic" are grouped in a facet called "type"; the dimensions "ascending" and "descending" are grouped in a facet called "direction"; the "sharp" and "flat" dimensions are grouped in a facet called "key signature". It can be seen at figure 6.5 that the key signature facet was separated in smaller dimensions: "0 to 2 sharps or flats", "3 to 5 sharps or flats", and "6 to 7 sharps or flats".

Writing the test

1- Obtain evaluation items: Taking into account facets, items that will be used to evaluate students are written. For example, three minor scales (A, B, and E) correspond to facets "key signature: 0 to 2 sharps", "type: harmonic" and "direction: ascending".

2- Numbering each item: A number, either attributed randomly or in an another way, is assigned to each of the question; for minor scales, there are sixty possible items (figure 6.5).

		TYPE			
		Harmonic		Melodic	
		DIRECTION		DIRECTION	
KEY SIGNATURE		ascend.	descend.	ascend.	descend.
Sharp	0 to 2	5- A 9- E 60- B	14- A 30- E 36- B	31- A 37- E 42- B	16- A 55- E 44- B
Sharp	3 to 5	2- F# 25- C# 41- G#	7- F# 18- C# 49- G#	4- F# 27- C# 35- G#	12- F# 52- C# 57- G#
Sharp	6 to 7	17- D# 21- A#	29- D# 43- A#	10- D# 51- A#	32- D# 38- A#
Flat	1 to 2	6- D 53- G	22- D 58- G	56- D 59- G	13- D 28- G
Flat	3 to 5	3- C 19- F 39- Bb	8- C 23- F 45- Bb	11- C 26- F 47- Bb	15- C 33- F 54- Bb
Flat	6 to 7	20- Eb 40- Ab	24- Eb 46- Ab	1- Eb 50- Ab	34- Eb 48- Ab

Figure 6.5 Resultant scheme.

Elaborating the correction tool

The aim is to produce several tables allowing the analysis of the student's results. The following components have to be included in each table (figure 6.6):

1. The facets and dimensions, and the number of each question;

2. the total number of questions used for each dimension;

3. the minimum number of questions that must be answered correctly. Mastery learning theory authors such as Block and Anderson (1975), and Bloom, Madaus and Hastings (1981), set this minimum to be 80 or 90 percent;

4. a cell to write the result of the student;

5. a cell to indicate whether a student has mastered the facet or set of facets or not.

The first table presented at figure 6.6 refers to a single facet; the second one refers to the mix of facets direction and type. Figure 6.5 illustrates using a circle, many incorrect answers that correspond to a problem that the second analysis table presented at figure 6.6 could precisely identify: that is a student that does not master the descending melodic minor scales (fourth row of the second table). Using the results of that analysis, it is possible to give him or her the relevant remedial teaching.

In the academic sector, these analysis tables can be used by a teacher to have a better understanding of the specific area that a student does not master. It is a decision tool aiming at proposing relevant remedial teaching when necessary. In regard of CODAMA, the recommendation to use the multiple facet scheme does not involve that tests will be developed. But, as the user's questions are the raw material of the user/CODAMA transactions, his or her requests and errors can be linked to the relevant cell of the scheme; this in turn allows the automatic production of the analysis tables. Then, CODAMA uses the different values recorded in this user model to take decisions in regard of the content of its next intervention(s). In fact, the multiple facet scheme and analysis tables associated can be used as a basis to produce the user model.

FACET TYPE	Questions	Minimum	Result	Mastered (X)
Harmonic Q- 2- 3- 5- 6- 7- 8- 9- 14- 17- 18- 19- 20- 21- 22- 23- etc.	30	24		
Melodic Q- 1- 4- 10- 11- 12- 13- 15- 16- 26- 27- etc.	30	24		

FACET DIRECTION/TYPE	Questions	Minimum	Result	Mastered (X)
Ascending/Harmonic Q- 2- 3- 5- 6- 9- 17- 19- 20- 21- 25- 29- 39- 40- 41- 60	15	12		
Ascending/Melodic Q- 1- 4- 10- 11- 26- 27- 31-37- etc.	15	12		
Descending/Harmonic Q- 7- 8- 14- 18- etc.	15	12		
Descending/Melodic Q- 12- 13- 15- 16- etc.	15	12		

FACET TYPE/BREAKING	Questions	Minimum	Result	Mastered (X)
Harmonic/Sharp Q- 2- 5- 7- 9- 14- 17- 21- 25- 29- 30- 36- 41- 43- etc.	16	13		
Melodic/Sharp Q- 4- 10- 12- 16- etc.	16	13		
Harmonic/Flat Q- 3- 6- 8- 19- etc.	14	11		
Melodic/Flat Q- 1- 11- 13- 15- etc.	14	11		

Figure 6.6 Examples of analysis tables.

When it is not possible to divide a task into small parts, another formative evaluation tool has to be used. We recommend the hierarchical test. Each question is linked to an objective being part of a hierarchy. It is a test centered on many objectives. The diagnostic interpretation is made by taking into account each part of the test; each part proposes several questions that relate to a particular objective (Scallon, 1988a). This allows the identification of what objective(s) a student does not master. As for multiple facet scheme analysis of results, a table is elaborated in order to be able to analyze the results obtained by a student. The main difference is the nature of the diagnosis. The one obtained when multiple facet sheme is used allows the identification of a precise difficulty in regard of a task having been divided into small parts; the diagnosis obtained when a hierarchical test is used refers to the identification of each intermediate objective that is not mastered. As example, the conceptual database modelling process being considered, it is possible that a user be not able to properly name entities.

Use of examples

According to the files issued from the previous step, some interventions have as component an example. Examples aim at stimulating the interest of the user; as mentioned by Robidas (1990), any individual recalls what concerns him or her, what is meaningful. To elaborate those examples, it is recommended to use real conceptual database models issued from historical files. As any organization has its own culture and vocabulary according to its mission (think about a bank versus an hospital), it has been decided that examples would be presented, when available, in regard of the type of organization being modelled. Because experts are different persons, expressing themselves in different ways, and moreover having no specialization in theory of instructional communication, the recommendation to use real conceptual database models issued from historical files involves that the structure of presentation of examples must be detailed. This structure being described, it ensures an unvarying formulation.

Means recommended to ensure unvarying formulation

Having reviewed literature related to instructional communication (Hartley 1985; Richaudeau, 1975, 1979; Rubinstein, and Hersh, 1984), we recommend that CODAMA's interventions that present examples be separated into two parts. The first part corresponds to the general statement of a definition, a principle, or a rule; consequently, it is a permanent part. The second part corresponds to an example; it is a variable part. When the statement of those examples is made at the next step, variables allowing to record data related to a particular case will be placed within the corresponding sentence; they refer to the data that will be collected from experts. It could be useful to automate the process of collecting the different values of the variables.

Taking into account the previous recommendations, the following knowledge acquisition method is proposed.

Knowledge acquisition method

Several tasks must be done and many refer to Gagné's model (1977), and Gagné and Briggs' principles of design (1979).

1. Analysis of content

1.1 Specialized manuals

Several manuals dealing with conceptual database modelling should be consulted to identify concepts, principles, and rules.

1.2 Syllabi

Syllabi obtained from several teachers and professors providing training to various types of learners (academic and professional) will be analyzed.

1.3 Course presentation

Courses given by teachers or professors in academic and professional environments will be observed and analyzed.

1.4 Synthesis to write unit objectives

The various performance unit objectives are written and the capability described by the action verb used in each statement is characterized in accordance with the taxonomy. It leads to a preliminary list of objectives.

1.5 Synthesis to write end objectives

The various performance end objectives corresponding to each unit objective are written and the capability described by the action verb used in each statement is characterized in accordance with the taxonomy. In order to accelerate the production of a first prototype which is needed to confirm the validity of this analysis, it is not necessary that all the performance unit objectives be divided into end objectives before moving on the following step.

1.6 Synthesis to write intermediate objectives

Here one writes the various performance intermediate objectives in regard of each end objective and characterizes the capability described by the action verb used in the statement in accordance with the taxonomy. In order to accelerate the production of a first prototype which is needed to confirm the validity of this analysis, it is not necessary that all the performance end objectives be divided into intermediate objectives before moving on the following step.

1.7 Elaboration of multiple facet schemes

The objectives for which it is possible to elaborate such a scheme are identified.

1.8 Formulation of topics associated to cells and set of cells

Explanations, keywords, questions, and rules are stated, and the user model is described.

1.8.1 Questions, rules and explanations

For each cell and set of cells, questions (passive and pedagogical resources), rules (active resource), and explanations related (active, passive and pedagogical resources), are written.

1.8.2 Identification of explanations presenting examples and statement of those examples

After having decided what explanations should propose an example, how they are stated is detailed.

1.8.3 List of keywords

A keyword is linked to each question, rule, and explanation, in accordance with each cell or group of cells.

1.8.4 User model description

Here, one describes which data are collected to produce the user model.

1.8.5 Fixing errors level allowed

Here, one has to fix what criteria CODAMA will use to infer that a user faces difficulties with a facet or a set of facets.

1.8.6 User model functioning

Here, one has to describe what happens during a work session in regard of data collected by CODAMA in view of being able to adapt its transactions of help to the user's needs.

1.9 Formulation of topics related to other objectives

Explanations, keywords, questions, and rules are stated, and the user model is described.

1.9.1 Questions, rules and explanations

For each objective, questions (passive and pedagogical resources), rules (active resource), and explanations related (active, passive and pedagogical resources), are stated.

1.9.2 Identification of explanations presenting examples and statement of examples

Here, one must determine what explanations will present examples. In order to be able to supervise experts, the statement of those examples has to be decided.

1.9.3 List of keywords

A keyword is linked to each question, rule, and explanation.

1.9.4 User model description

Here, one must describe which data will be collected in regard of the user model.

1.9.5 Fixing errors level allowed

Criteria that CODAMA will use to infer that a user does not master an objective or a set of objectives are determined.

1.9.6 User model functioning

Here, one has to describe what happens during a work session. Recorded data allowing CODAMA to be able to adapt its transactions of help to the user's needs are described.

1.10 Keywords network elaboration

Links between keywords are illustrated by a network.

1.11 Hierarchy with intervention/objective links

Hierarchical links existing between interventions and objectives are illustrated.

2. Use of content analysis

2.1 Users performing task

Here, one has to observe and film (if possible) users elaborating a conceptual database model, and to ask them to formulate their requests for help out load.

2.2 Nature of difficulties

The list of questions and rules elaborated during the analysis of content is used to point out questions and problems encountered by users. Frequency of requests for help will also be indicated. If some aspects are missing, they are indicated.

2.3 Adjusting content

Depending upon results obtained at 2.2, some questions, rules and explanations are either added, reformulated or deleted. All related elements, such as facets, hierarchy, keywords, networks, are adjusted.

CHAPTER SEVEN

CASE STUDY 2

MUSICAL COMPOSITION ADVISOR SYSTEM: FILES RELATED TO THE DESCRIPTION OF THE KNOWLEDGE ACQUISITION STRATEGY

Main aspects files of step one treated of

- Objectives are written (Secondary music program)
- Statements of activities are written
- Scholastic context
- Access to interventions (prerequisites) by selecting keywords
- MUSIC must be able to do comments on compositions
- MUSIC must be able to compose

Scholastic context

The various levels of objectives stated in the secondary level music program underlie the acceptance of the cumulative learning hypothesis. Several theories and models were developed in regard of this hypothesis. One of those was elaborated by Bruner (1962, 1963, 1966a, 1966b). His theory and the model associated were in the past used to develop the Manhattanville Music Curriculum Program (s.d.). The following is an example of spiral curriculum: First musical composition activities may be developed in regard of bars having as time unit the quarter.

Time signature of the melodies can be either $\frac{2}{4}$, $\frac{3}{4}$, or $\frac{4}{4}$. The advisor having been used several times, bars having as time unit the half can be added. Later, those having as time unit the eighth can also be added. In the same way, the student is limitated in regard of note and rest signs he or she can use; for example, at first, the student is allowed to use only halfs, quarters, and eights. Later, sixteenths are added. Rest signs associated are afterwards added. First melodies refer merely to Major keys. Later, minor keys are added, etc.

Description of means recommended to take into account the scholastic context

The spiral curriculum proposed by Bruner (1962, 1963, 1966a, 1966b) should be used to plan the advance of musical composition activities. As a theory and a concrete example of application to music teaching do exist, works to be done by individuals in charge of the elaboration of the content will be easier. However, mechanisms allowing music teacher to modify the advance of activities must be developed.

Providing access to prerequisites by selecting keywords

Files issued from the previous step mention that access to prerequisites, and to links between them, will be ensured by outlining keywords within each explanation.

Description of means recommended to provide access to prerequisites by selecting keywords

We recommand that a network be elaborated in order to illustrate what concepts, principles, and rules, are referred to when a particular explanation is presented.

To do comments and compose

Files stemming from step one mention that MUSIC has to be able to do comments on each student's composition. It is also stated that it has to be able to compose a melody, and to do comment on it.

Means recommended to allow MUSIC to do comment on a composition and compose

In regard of rules to be used to analyze or compose a melody, it is recommended to formulate them in terms of determinant elements. Two types of determinant elements have to be defined: those that must be enforced, and those that can be used. An example of determinant element that must be enforced in regard of the question-answer musical composition process relates to the number of bars: the number of bars of the answer must be equal to the question one. An example of determinant element that can be used relates to the relation between the question and the answer; it can be parallel, semi-parallel, or contrasting.

Sources of information

To elaborate interventions, it is recommended to use the various documents produced in regard of the secondary level music program (1981b, 1983), the Manhattanville Music Curriculum Program (s.d.), and specialized books.

It is recommended to refer to specialized books addressing the topic of musical composition to produce the list of determinant elements. These elements being used, MUSIC will be able to compose, and to do comments either on its own composition, or on the student's one. In regard of the use of determinant elements, it is recommended that students perform on paper musical composition activities for which MUSIC must intervene. Music teachers and professional composers will then do comments on these melodies. It will allow the determination of the most frequent weaknesses in regard of the transfer of knowledge; it will also provide many samples of experts' comments aiming at helping each student to make progress. Links with prerequisites must be outlined.

Knowledge acquisition method recommended

1. Analysis of content

1.1 Prerequisites identification

A tentative list of prerequisites will be produced using both the list of objectives and content associated presented in the documents related to the secondary level music program, and the statements of musical composition activities. Links between prerequisites and objectives of the music program have to be defined.

1.2 Consultation of specialized books

Two categories of books should be analyzed: those laying out concepts, principles, and rules related to musical language in general, and those setting out particularly musical composition.

1.3 Synthesis in view of formulating the tentative content of explanations

Using results obtained at 1.1 and 1.2, and taking into account the future user of MUSIC, a first version of the content of explanations will be elaborated.

1.4 Links between prerequisites

The content of each explanation will be analyzed in order to be able to determine to what other prerequisites it refers.

A network will be produced. Vocabulary used will be made uniform.

1.5 Analysis of students' compositions

Some students will do compositions. This will allow the identification of their weaknesses, and their needs for help. Their compositions will after be analyzed by music teachers and professional composers. They will do comments on them.

1.6 Consultation of specialized books related to composition process

Those books will be analyzed in order to make out a list of determinant elements of a composition in regard of the musical composition processes that activities proposed to the students refer to.

1.7 Synthesis to make out a list of determinant elements and identify comments related

Results obtained from 1.3, 1.5 and 1.6 will allow the production of a sole list of determinant elements, and the elaboration of rules associated. Comments related to the observation of the presence or the absence of each element will also be written.

1.8 Sequencing the activities in accordance with the spiral curriculum

Progression of activities will be planned according to the academic year. To do so, secondary level music program and Manhattanville Music Curriculum Program will be used.

CHAPTER EIGHT

STEP 3. DESCRIPTION OF THE CONTENT, THE ARCHITECTURE, AND OF THE IMPLEMENTATION IN THE ENVIRONMENT STRATEGY

The tasks to be done at this step in regard of developing the content refer to the application of the knowledge acquisition method being described in files of step two. If some specifications cannot be fulfilled, there will be a look back; therefore, some parts of step one, or two, or both, will have to be done again.

Sources of knowledge

In accordance with the means described in the knowledge acquisition method, sources of knowledge identified are analyzed.

Statement of the interventions

The statement of each intervention is written in accordance with the knowledge acquisition method.

Architecture of the advisor

The container being considered, the advisor's structure is outlined, without taking physical aspects into account. Global characteristics of the advisor are illustrated and documented. The main components are identified. The identification of these main components brings about the identification of the main modules, and of links between them. To elaborate models of data and treatments, one can decide to use conventional techniques such as structured analysis for treatments (data flow diagrams), and conceptual database modelling for data (entity-relationship diagram); object oriented analysis can also be used. The main point is to not think about physical aspects.

When using systems analysis and techniques, one of the key philosophies that must be understood is the difference between logical and physical aspects. For example, data flow diagrams are used to produce a model of an advisor that shows the data flows between processing tasks, and the various points at which data is stored, retrieved, or both. The starting point is the context diagram; it aims at showing the highest level model of the advisor. Its purpose is to identify what is to be included in the area under study. It helps to illustrate the link of the advisor with the rest of the world. It allows to obtain a broad overview of what the advisor encompasses and what it does not encompass. A context diagram being elaborated, one can make data flow diagrams from it. It is a graphic representation of an advisor that shows data flows to, from, and within the advisor, processing functions that change the data in some manner, and the storage of this data. It is a sort of network of related advisor functions indicating from where information are received and to where it is sent. This graphic representation facilitates any modification that could be done for different reasons such as adjusting the way a resource interacts with the users, after a prototype has been tried out.

Documentation of the architecture

While doing the analysis, analysts have to document it. A data dictionary is used. It is documentation that supports diagrams. For example, data flow diagrams being considered, it contains all the terms with their definitions used for data flows and data stores that relate to a specific advisor. The purpose of such a dictionary is to define the contents of the data flows and data stores. Data dictionary is necessary to provide consistency.

In the following sections, two approaches are shortly described. They are used in the case studies. One of these approaches is the Gane and Sarson's (1979) one; it aims at producing data flow diagrams. The second is the entity-relationship approach (Chen, 1976): it aims at producing entity-relationship diagram. The Merise formalism (Tardieu, Rochfield, and Coletti, 1983) is used. Note that because we don't refer to them in the case studies, other approaches to process and data analysis, and object-oriented approaches, are not described here.

The Gane/Sarson approach

According to Gane and Sarson (1979), the purpose of logical modelling is to take ideas about requirements and convert them into precise definitions as fast as possible. Requirements of the advisor have been carefully planned and detailed before the process of logical modelling begins. To convert them into precise definitions, Gane and Sarson (1979) propose graphical techniques that enable an analyst to put down the essence of a system without going through the trouble of actually physically implementing it.

At first, analysts have to develop a system-wide dataflow diagram (DFD) describing the underlying nature of what occurs in the advisor. This DFD achieves three things. First, it sets a boundary to the area of the advisor. Second, it is nontechnical. Nothing is shown on a DFD that is not easily understandable to people familiar with the advisor depicted, whether or not they know anything about computers. Third, it shows both the data stored in the advisor and the processes that transform the data. It shows the relationships between the data and the processes in the advisor.

Four symbols are used to produce a picture of the underlying nature of any system, at any desired level of detail. Symbols are illustrated at figure 8.1: note the box for external entity, the rounded box for the process, the data-flow arrow, and the open rectangle for data store. Things represented by the external entity symbol are, by definition, outside the system. Notice also that time is not shown on a DFD.

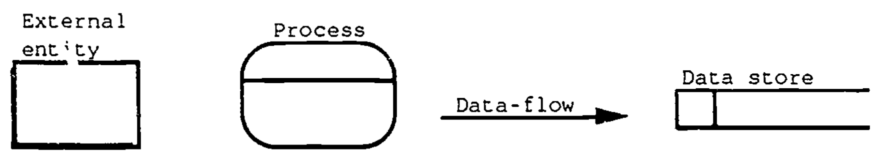


Figure 8.1 DFD symbols.

An advisor-wide DFD being developed, analysts derive a first-cut data model, i.e., a list of the data elements to be stored in each data store, as defined on the DFD.

Third, analysts see what entity-relationship analysis (described in the next section) can tell about the structure of the data to be stored in the advisor. In fact, they ask them the following questions: What are the entities of interest about which data may need to be stored? What, if any, relationship exist between a pair of entities?

Using all the information they have about the data, analysts describe the data model. After that, they redraft the DFD to reflect a more precise view of the advisor as a result of entity-relationship analysis.

The two next steps deal with converting the logical model into a physical model. They will be summarized at chapter eleven.

The entity-relationship approach

As stated by Chen (1976), and Tardieu, Nanci and Pascot (1984), the entity-relationship approach is a methodology aiming at systematically converting user requirements into well-designed databases. It makes the integration of different application systems easy. The entity-relationship methodology is simple, easy to understand by noncomputer people, and theoretically sound.

At first, analysts identify entity-relationship types and associated properties and also the identifier for each entity type.

An entity is a concept, an event, an organization, a person, or a thing of interest to the organization doing the modelling. An entity type is a classification of entities satisfying certain criteria. A relationship is an interaction between entities. A relationship type is a classification of relationships based on certain criteria. Figure 3.1 depicted the MERISE entity-relationship formalism (Tardieu, Nanci, and Pascot, 1984) in which rectangular boxes represent entity types and ellipses represent relationship types.

Entities and relationships being identified, the next step is to identify the cardinality of the relationship types. There are minimum cardinalities and maximum cardinalities. The minimum cardinality is the minimum number of times occurrences of an entity type are involved in a relationship type. Minimum cardinalities allowed are 0, 1, 1, c, and n (Note: 1 corresponds to an encircled cardinality according to Merise formalism; it aims at declaring an entity as being dependent on another one; it modifies the identifier); each has a different meaning. The maximum cardinality is the maximum number of times occurrences of an entity type are involved in a relationship type; maximum cardinalities allowed are 1, 1, c, and n; each represents a different reality.

The entity-relationship diagram being elaborated, analysts convert it either in conventional file and database structures, in statement of facts, predicates, rules, or in another way. This decision refers to step four which will be described at chapter eleven.

Strategy of implementation in the environment

While analysts elaborate logical models, other members of the team detail the elements of the strategy of implementation of the advisor in the environment. Despite the strategy of intervention of the advisor has been detailed at step one, the sources and means used to do the knowledge acquisition has been selected at step two, and the knowlegde acquisition has been done at the beginning of the current step, the particularly complex problem to develop an advisor is not totally solved.

Works done focused on one aspect of the problem, being the transmission of the knowledge in a help context. Modelling tasks currently done focus on another aspect, being to find the best technological solution for the transmission of knowledge in regard of this particular context. Simultaneously, works related to another aspect begin; it is the problem of implementing in the environment the technological solution.

To describe the strategy of implementation of the advisor in the environment is in a way to describe how the marketing of this instructional technology will be done. Unfortunately, that is an aspect that is not abundantly documented in the literature. This particular problem was not really addressed by the mean of experiments rigorously designed. That is another particularly complex problem that has been, in the past, transformed in a simplified reality. In other words, because it was observed that a proportion X of individuals faced transfer of knowledge problems while performing complex tasks, because it was decided to propose a technological solution to manage these problems, because estimation of the utility of the intelligent advisor technology has shown that it would favor the transfer of knowledge for Y% of the proportion X of individuals, one tends to postulate that the advisor will actually be used, thus useful. It is a simplified vision of the reality.

To be useful, an advisor must be used. To ensure that an event will occur, it must be managed. To be able to manage, all the events that can prevent a desired situation to occur must be identified; in the same way, all the events that can cause that the desired situation no longer occur at a certain moment in the time must be identified. The desired situation is that the advisor be used at the beginning of its implantation in the environment and after, even if some changes occur in the environment. Employees retiring is an example of change in the environment. They are replaced by other persons. These persons were not part of the environment when the advisor was implemented. Because this is an event that can affect the use of the advisor, means to ensure that these new employees will have all the needed informations in regard of the advisor must be planned. The more the events are forecasted, the more they can be managed; to manage them, a strategy of implementation in the environment is elaborated.

But, as mentioned before, the identification of factors and events that ensure that the implementation of an instructional technology will be successful is not a documented topic. Perhaps because this so important topic is not a visible part: it is difficult to plan a demonstration aiming at showing a strategy of implementation. Although, several questions arise.

One of these questions relates to the proper moment to introduce the instructional technology in the environment. This proper moment having been found (or guessed, because the current state of knowledge of this domain), another problem is to determine how it will be introduced. Will the target users be informed in advance? If the answer is yes, two other questions arise: How long in advance? How many announcements will be done? No matter the tactic selected, i.e., either deciding to wait the actual moment of installation, to announce one time one month before the installation, or to announce several times for a period of six months, the content of communications must be carefully detailed. When many communications are planned, the content and the sequence must be detailed.

In regard of the content of communications, there are several possibilities. It can be decided to describe the tool and its limits. That means the worth of the instructional technology, an intelligent advisor system in our case, will not be magnified. Any technology does not transform a complex transfer task in an easy one. As example, we will refer to a spreadsheet being used to do calculations related to the real estate domain. It does not make easier the perceiving and thinking processes. To find what is the best real estate investment for an organization at a certain moment in the time, it is to make hypothesis about the future and to take into account a current socioeconomic context. The use of a spreadsheet accelerates the process of the visible part of the problem, i.e., the numbers. An intelligent advisor that would be interfaced to a spreadsheet to help a user facing such a complex problem could help to perceive and think; however, it could not simplify the problem.

Another possibility is the illusion tactic. It is to organize a demonstration or to describe its functioning on paper, or both, taking care to present only a sample of simple cases of transfer tasks. It can convince the target users for a time that it will make easier their works. As long as the high level managers do not decide to enrich their tasks, this tactic can contribute to enhance the motivation of the target users to use the technology.

Using memo announcing that an intelligent advisor is available, or has been or will be installed, is another possibility. The name and phone of the person to reach to have more information are mentioned. This tactic was used in the past for the scholastic environment.

Mechanisms allowing to periodically recall that an instructional technology is available can also be planned. The difficulty is to find the proper frequency. To find it, data related to changes in staff for which the technology was developed can be used.

Once more, remind that the development life cycle is user oriented. The strategy of implementation of the advisor in its environment is detailed in order to ensure that <u>all</u> the target users will <u>always</u> have <u>all</u> the information they need, no matter changes that occurred in the organization. An example of change that can cause lost of informations is an employee that quits; it is not sure that the information transmitted in order to prepare the environment to the installation of an instructional technology, i.e. aim, type of help, limits, etc., will be available to the new person hired. To be sure that <u>everybody</u> <u>always</u> have <u>all</u> the information needed to use the advisor, or <u>always</u> have an easy and quick access to this information, whatever changes occur in the organization, is to have a detailed plan of communication for <u>all</u> the life of the advisor.

It also means that a person must be permanently and actually affected to the management of the post implementation[1].

Table 8.1 summarizes the main decisions to be taken during the third step of the development life cycle.

Table 8.1

Summary of the third step

1.	To analyze sources of content.
1.x	*Steps of the knowledge acquisition method.*
2.	To elaborate the content of interventions.
2.x	*Steps of the knowledge acquisition method.*
3.	To produce the architecture of the advisor.
3.x	*Steps of the systems analysis method selected.*
4.	To document the architecture.
4.x	*Format of the systems analysis method selected.*
5.	To describe the environment implementation strategy.

Iteration

Tasks related to this step inevitably generate many looks back to previous steps. Developing an advisor is a very complex problem, so it is impossible to design and develop the best possible solution on the first attempt. Through iteration, the advisor is gradually refined on. Diagrams developed at this step are easy to revise, so they facilitate the iteration.

[1] The strategy of implementation of the advisor in the environment is not presented in the case studies.

CHAPTER NINE

CASE STUDY 1

CONCEPTUAL DATABASE MODELLING ADVISOR: FILES RELATED TO THE DESCRIPTION OF THE CONTENT AND OF THE ARCHITECTURE[1]

1. Analysis of content

1.1 Specialized manuals

More than twenty books have been analyzed. The list is presented in the real files (Boulet, 1988a, 1988b, 1988c; Boulet, and Barbeau, 1988).

1.2 Syllabi

Ten syllabi have been analyzed; five of those issued from academic sector, and five others, from professional sector. They are presented in the real files.

1.3 Course presentation

Two courses (45 hours each), and a one day training session given within an organization have been observed.

---

[1] Note: As this chapter aims at putting in concrete form notions presented in previous chapters, only a subset of the content is included.

1.4 to 1.6 Synthesis to write objectives

The domain has been analyzed in order to identify the prerequisites (more details can be found in: Boulet, 1988a, 1988b, 1988c; Boulet, and Barbeau, 1988). The aim of CODAMA being "to help the user of a graphics software to elaborate a conceptual database model" has been divided into four unit objectives (U.O.x). Each of these unit objectives has been divided into terminal objectives (T.x). Intermediate objectives (I.x) associated to each terminal objective were then identified. The statement of objectives is presented at tables 9.1, 9.2, 9.3, and 9.4.

Table 9.1

Unit objective 1

U.O.1. To start the process of designing a conceptual database model (Higher-order rule)
T.1: To explain what is a conceptual database model (Set of propositions)
I.1: To define conceptual database model (Idea)
I.2: To state the philosophy underlying the conceptual database modelling (Proposition)
I.3: To describe the role of the conceptual database model (Idea)
I.4: To enumerate the elements of a conceptual database model (Proposition)
T.2: To make up a list of properties (Rule)
I.1: To select the information that has to be represented in a conceptual database model (Rule)
I.2: To define property (Idea)
I.3: To define occurrence of property (Idea)
I.4: To identify properties (Rule)
I.5: To explain where to find properties (Set of propositions)

Table 9.2

Unit objective 2

U.O.2. To identify and represent the entities, their identifiers, and their properties (Higher-order rule)

T.1: To explain the concept of entity and its characteristics using examples (Concept)

- I.1: To define entity (Idea)
- I.2: To describe the characteristics of entities (Proposition)
- I.3: To give satisfactory names to the entities (Rule)
- I.4: To find occurrences of the entities (Rule)

T.2: To find an appropriate identifier for each entity (Rule)

- I.1: To define identifier of an entity (Idea)
- I.2: To explain the characteristics of the identifier of an entity (Proposition)
- I.3: To set the identifier, after finding an entity and its list of properties (Rule)

T.3: To find the entities, their identifiers, and their lists of properties, starting with the list of properties (Rule)

- I.1: To describe how to find entities, using an example (Set of propositions)
- I.2: A situation being given, to describe how entities, their identifiers, and their lists of properties were found (Rule)

T.4: To represent the entities, their identifiers, and their properties (Rule)

- I.1: To explain how to represent entities (Set of propositions)
- I.2: An entity, its identifier, and its list of properties being given, to represent it (Rule)

Table 9.3

Unit objective 3

U.O.3. To identify and represent the relationships of a conceptual database model, their identifiers, and their properties (Higher-order rule)

T.1: To explain the concept of relationship and its characteristics using examples (Concept)

- I.1: To define relationship (Idea)
- I.2: To explain the characteristics of relationships (Proposition)
- I.3: To give satisfactory names to the relationships (Rule)
- I.4: To find occurrences of the relationships (Rule)

T.2: To find an appropriate identifier for each relationship (Rule)

- I.1: To define identifier of a relationship (Idea)
- I.2: To explain the characteristics of the identifier of a relationship (Proposition)
- I.3: A relationship connecting two entities being found, to find the identifier of the relationship (Rule)

T.3: To find the relationships, their identifiers, and their lists of properties, starting with a list of properties (Rule)

- I.1: To describe how to find relationships, using an example (Set of propositions)
- I.2: To describe how to find relationships, their identifiers, and their lists of properties for a given situation (Rule)

T.4: To represent the relationships, their identifiers, and their properties (Rule)

- I.1: To explain how to represent relationships, using Merise formalism (Set of propositions)
- I.2: A relationship and its list of properties being given, to represent it (Rule)

Table 9.4

Unit objective 4

<u>U.O.4. To set the cardinalities and constraints</u> (Higher-order rule)
T.1: For a given situation, to set cardinalities (Higher-order rule)
I.1: To define the concept of cardinality (Idea) I.2: To explain the role of minimum cardinality (Proposition) I.3: To explain the role of maximum cardinality (Proposition) I.4: To explain the role of cardinalities (Proposition) I.5: To present rules applied in writing the cardinalities (Proposition) I.6: To explain the context of use of each individual cardinality (Set of propositions) I.7: To make the proper association between a context and a minimum cardinality (Rule) I.8: To make the proper association between a context and a maximum cardinality (Rule) I.9: To explain the context of use of each pair of cardinalities (Set of propositions) I.10: To make the proper association between a context and a pair of cardinalities (Rule) I.11: To explain the context of use of each couple of pairs of cardinalities (Set of propositions) I.12: To make the proper association between a context and a couple of pairs of cardinalities (Rule) I.13: To illustrate with examples occurrences of couples of pairs of cardinalities (Rule)
T.2: For a given situation, to set constraints of integrity (Higher-order rule)
I.1: To define the concept of constraint of integrity (Idea) I.2: To illustrate with an example a constraint of integrity (Rule) I.3: For a given situation, to determine whether there is a constraint of integrity (Rule) I.4: To represent a constraint of integrity (Rule)

1.7 Elaboration of multiple facet schemes

In regard of unit objective 4, some multiple facet schemes were elaborated. The scheme illustrated at figure 9.1 relates to many objectives. Cells located under the heading minimum refer to U.O.4/T.1/I.2, U.O.4/T.1/I.6, and U.O.4/T.1/I.7 (The statement of these objectives was presented at section 1.6); it can be seen that minimum cardinalities allowed are 0, 1, $\underline{1}$, c, and n (Note: According to Merise formalism, $\underline{1}$ corresponds to a cardinality being encircled). Cells located besides the heading maximum correspond to U.O.4/T.1/I.3, U.O.4/T.1/I.6, and U.O.4/T.1/I.8; it can be seen that maximum cardinalities allowed are 1, $\underline{1}$, c, and n. The total number of pairs used can be found by looking up each intersection; these data are more specifically referring to U.O.4/T.1/I.9 and U.O.4/T.1/I.10. Empty cells represent interventions related to pairs of cardinalities. Cells with a x represent errors; it is incorrect to use such pairs. They relate to interventions of the active resource.

		Minimum				
		0	1	1̲	c	n
M	1			x	x	x
a	1̲		x		x	x
x	c			x		x
i	n			x		

Figure 9.1 Multiple facet scheme representing individual cardinalities and pairs.

Another scheme (figure 9.2) represents the couples of pairs of cardinalities; it refers to U.O.4/T.1/I.11, U.O.4/T.1/I.12, and U.O.4/T.1/I.13. This scheme can also be used, when necessary, to calculate the number of pairs used.

		Right leg									
		0,n	1,n	1,1	1,1	0,1	0,c	1,c	c.n	c,c	n,n
Left leg	0,n										
	1.n										
	1,1										
	1,1										
	0.1										
	0,c										
	1,c										
	c.n										
	c,c										
	n,n										

Figure 9.2 Multiple facet scheme referring to couples of pairs of cardinalities.

1.8 Formulation of topics associated to cells and set of cells

1.8.1 Questions, rules, and explanations related

For each cell or group of cells, questions (passive and pedagogical resources), rules (active resource) and explanations related are formulated[2]. The number associated is only an identifier; it is not displayed.

<u>Pedagogical resource</u>

Questions

Table 9.5 presents the list of questions. In regard of variables, X can have the values 0, 1, $\underline{1}$, n, and c; Y can have the values 1, $\underline{1}$, n, and c.

Table 9.5

Questions related to facets

How do we define the concept of cardinality?(11110)
What is a minimum cardinality?(11120)
What is a maximum cardinality?(11130)
When do we use cardinalities?(11140)
How do we represent cardinalities?(11145)
What does mean a cardinality X?(11150 to 11180)
When do we set cardinalities? (11190)
How important is it to use one specific pair of cardinalities instead of one another? (12101)
Can we have a pair of cardinalities changing in function of time for two same entities? (12102)
What does mean a pair of cardinalities X,Y?(12103 to 111)
What does mean a couple of pairs of cardinalities X1,Y1-X2,Y2?(12112 to 12144)
Can we combine all pairs of cardinalities? (12145)

[2] For the purpose of this book only a subset of explanations, questions, and rules, will be presented. More details can be found in Boulet and Barbeau (1988).

Explanations related

The content of explanation 11150 is formulated as follows:

11150: A cardinality of 0 means that **entity** (Q 31110) X can exist by itself, without **occurring in a relationship** (Q 41118). The value 0 can be assigned only to minimum cardinalities; otherwise, the entity X would never exist in the conceptual database model.

EXAMPLE: Being an organization, owner of a building, aiming at managing its parking; it needs to know which employees own a car.

Example of form used: (not depicted here)

Form Employee
Employee number
Employee name
Employee first name
Employee address
Employee car registration plate number

Entities Employee and Car, and relationship Owns can be identified. An Employee does exist for the organization even if he does not Own a Car, in which case the entity Employee would have the minimum cardinality of 0 with respect to the relationship Owns.

Passive resource

Questions

The list presented here would be displayed if a user selected an entity named Student being in relationship Enrolls with another entity Course; the couple of pairs of cardinalities is 0,n-0,n.

11210 Explain minimum cardinality 0 of Course?
11215 Explain maximum cardinality n of Course?
11220 Explain minimum cardinality 0 of Student?
11225 Explain maximum cardinality n of Student?
11230 Explain pair 0,n of Course?
11240 Explain pair 0,n of Student?
11250 Explain the couple 0,n-0,n?

Explanations related

The explanation related to item 11210 is formulated as follows:

11210 Minimum cardinality of 0 of **entity** (Q 31110) Course means that a Course does exist even if it is not given during a trimester; so, it may happen that no Student is Enrolled. In the present case, is it true that entity Course does exist without having to participate to the relationship Enrolls?

Active resource

Rules

CODAMA does many checks. Two examples are:

11285 Minimum cardinality greater than maximum cardinality on a same leg.
11290 Maximum cardinality equal to 0.

As it was not possible to find any historical data being reliable and valid in regard of the frequency of use of each individual cardinality, pair or couple, rules allowing comparison were not defined.

Explanations related

A single example is presented here.

11285 **Minimum cardinality** (Q 11120) X of the **entity** (Q 31110) *<name of the entity being used in the model>* has a value greater than **maximum cardinality** (Q 11130) Y.

1.8.2 Identification of explanations presenting examples, and statement of those examples

Explanations 11145, 11150 to 11180, and 12103 to 12144 propose examples. The organizational context is first presented: kind of organization, kind of clients, and kind of transactions. In the following example, what an expert described in regard of item 11160 is indicated between <> (minimum cardinality 1).

Given a <manufacture of furniture>. Its clients are <retail shops, persons, and organizations>. This enterprise <makes to order>. When it receives an <order>, the following form is used:

<Order (not illustrated)

Order number
Customer number
Customer name
Customer address
Furniture number
Quantity
Order date
Delivery date
Unit price>

Entities (Q 31110) <Order> and **relationship** (Q 41110) <Concerns> can be identified. An <Order>, to exist in the memory of the organization, must <Concern> at least one <Furniture>. A minimum cardinality of 1 is placed for entity <Order> in regard of the relationship <Concerns>.

1.8.3 List of keywords

Examples of explanations were presented at sections 1.8.1 and 1.8.2. In those explanations, bold characters are used for some words that are followed by a number between parenthesis (Such as **entity** (Q 31110), **relationship** (Q 41110)); they are keywords. In the real files, the complete list is presented.

1.8.4 User model description

Figure 9.1 (Section 1.7) presents the scheme related to individual cardinalities. Data related to the frequency of use of each individual cardinality are placed in the corresponding cell. The total number of individual cardinalities used may be found by adding up the values of the table. The total number of minimum cardinality may be found by adding up each column. The total number of maximum cardinality may be found by adding up each row. The total number of each pair of cardinalities may be found by looking at each intersection. Couple of pairs of cardinalities is represented by scheme presented at figure 9.2 (Section 1.7). The total number of each couple of pairs of cardinalities may be found by looking at each intersection. The total number of each pair used may be found by adding up row and column corresponding to a pair. The total number of couples may be found by adding up the values of the table.

To keep a trace of consultations made by the user, either by menus or selection of keywords, and of errors, CODAMA will elaborate two similar tables. Using these data, it decides what information will be presented. Order of consultations and time will also be recorded.

1.8.5 Fixing errors level allowed

The number of consultations allowed will be calculated in accordance with mastery learning theory (Bloom, 1974, 1979). A student has to succeed at least 80% of items related to a formative test centered on one objective (Caponigri, 1980); a student may fail about 20% of items. In the same way, a user will be allowed to have access to an explanation related to the same intermediate objective until he reaches 20% of the number of elements used in his or her conceptual database model (Such as 20% of minimum cardinality X).

1.8.6 User model functioning

Since it would not be useful to explain all the changes that happen in the numerous tables, we shall use a simplified example. Suppose a person is using a graphics software to develop a conceptual database model. Further suppose that this user has chosen CODAMA's passive resource (more specifically, the user questions CODAMA about cardinalities). The user selected entities Student and Course, and the relationship Enrolls. The couple of pairs of cardinalities is 0,n-0,n. The menu is the one that was presented at section 1.8.1. As the user progresses, several tables are elaborated. We will illustrate parts of these tables.

First, CODAMA elaborates the table presented at figure 9.2; it computes the total number of couples of pairs of cardinalities (Figure 9.3 presents a subset of the table). For the purpose of the example, let's suppose there are 78 (Note: we suppose that other values are placed in cells not illustrated at figure 9.3).

Pair of cardinalities (right leg)

		0,n	1,n	1,1	(1,1)	0,1	
	0,n		12	16			
pair	1,n		20	12			
of)	1,1						

Figure 9.3 Part of a table related to couples.

Another table is elaborated in order to determine the number of consultations allowed for each couple in accordance with the 80% success level (figure 9.4).

Pair of cardinalities (right leg)

		0,n	1,n	1,1	(1,1)	0,1	
	0,n		3	3			
pair	1,n		4	3			
of	1,1						

Figure 9.4 Part of the table related to consultations allowed.

The table illustrated at figure 9.1 is also elaborated.

It can be seen at figure 9.3 that there are 12 couples 1,n-1,1. Thus, the user can make 3 consultations in regard of this couple; after 3 consultations, the item allowing to have explanation according to this couple will not be displayed in the menu. The total number of pairs will be calculated by using data of the table presented at figure 9.1.

If the user asks for the first time to explain a couple 1,n-1,1, tables representing the user model will be updated; an example is presented at figure 9.5. Note that only the table related to couple is presented.

Pair of cardinalities (right leg)

		0,n	1,n	1,1	(1,1)	0,1	
	0,n		0	0			
pair	1,n		0	1			
of	1,1						

Figure 9.5 Part of the user model related to couple.

CODAMA realizes that only one consultation has been made in regard of couples (sum of the table), and of couple 1,n-1,1 (value of intersection 1,n-1,1).

Because this is the first time the user has ref' red to this piece of knowledge, the user's question will be answered. But now suppose that the user asks three more questions about three 1,n-1,1 couples. At this point, it becomes questionable how well the user understands the couple in question. CODAMA updates its model of the user as illustrated at figure 9.6.

Pair of cardinalities (right leg)

		0,n	1,n	1,1	(1,1)	0,1	
	0,n		0	0			
pair	1,n		0	4			
of	1,1						

Figure 9.6 Updating the table.

CODAMA knows that the number of queries made (4) is greater than the number allowed (3) for the couple 1,n-1,1; it also knows that the highest level ceiling (16) has not been reached. In the same way, it checks if other ceilings have been reached. If not, CODAMA takes the decision to block the access to the item related to couple, while keeping displayed the others allowing to have an access to explanations of pairs, and individual cardinalities; in the present example, these explanations are in regard of pair of cardinalities 1,n and pair 1,1, and minimum cardinality 1 of an entity X, minimum cardinality 1 of entity Y, maximum cardinality n of the entity X, and maximum cardinality 1 of the other one. That corresponds to lower levels of the hierarchy. According to the results of the analysis of content, to be able to understand concepts, principles, and rules related to couple of pairs of cardinalities, an individual must be able to use those related to pair of cardinalities. Therefore, CODAMA infers a knowledge user level. Note that as the user progresses, other tables are elaborated, updated, and taken into account while CODAMA makes hypothesis on the user knowledge level. The time between each consultation is also taken into account; it is part of the inference process.

The user may also decide to select ke''words. CODAMA updates the corresponding tables, ckecks their values, makes hypothesis on the user knowledge level, and displays explanations requested and relevant.

1.9 Formulation of the topics related to other objectives

1.9.1 Questions, rules, and explanations related

Questions (passive and pedagogical resources), rules (active resource), and explanations related are formulated[3]. Numbers are identifiers; they are not displayed to the user.

---

[3] For the purpose of this book only a subset of explanations, questions, and rules, will be presented. More details can be found in Boulet (1988a, 1988b, 1988c), and Boulet and Barbeau (1988).

Interventions related to unit objective 1

Pedagogical resource

Questions

Table 9.6 presents a part of questions related.

Table 9.6

Part of questions related to unit objective 1

What is a conceptual database model? (21110)
What is the philosophy that underlies the conceptual database modelling process? (21120)
What are the main elements of a conceptual database model? (21140)
What information should be represented in a conceptual database model? (21150)
What is a property? (21160)
What are properties used for? (21165)
What is an occurrence of property? (21170)
Can the occurrences of a property have different values? (21177)
How can you find properties? (21180)
Where can you find properties? (21190)

Explanations related

One example of explanation is presented here.

21110: A conceptual database model is a map, a diagram of an organization's memory. It is like a picture of its memory that was taken. This diagram represents in symbolic form all the information the organization needs in order to function, and shows how the bits of information are connected. Conceptual database models are concerned with data means. In the diagram, identifiable items called **entities** (Q 31110), and links between these identifiable items called **relationships** (Q 41110), are shown. Entities and relationships are described by **properties** (Q 21160). **Cardinalities** (Q 11110) are used to define the meaning of relationships.

Passive resource

Questions

A single item of the menu is presented here.

21220: What does property *<here the name of the property selected by the user is displayed>* mean?

BEST COPY AVAILABLE

Explanations related

The explanation related to item 21220 is:

21220: Property *<here the name of the property selected by the user is displayed>* means that the reality of the organization being observed, you conclude that *<here the name of the property selected by the user is displayed>* was a **primary information** (Q 21150), i.e., an element of information not described by smaller elements of information. Is it what you perceived?

Active resource

Rules

An example of rule is:

21310: Two properties or more have the same name

Explanations related

The explanation related to rule 21310 is:

21310: x properties have the same name *<name of the properties>*. To avoid confusion, the **name of the property** (Q 21180) should state clearly what is being referred to.

Interventions related to unit objective 2

Pedagogical resource

Questions

Table 9.7 presents a part of the items related.

Table 9.7

Part of questions related to unit objective 2

What is an entity? (31110)
What characteristics do entities have? (31112)
How do you name entities? (31114)
Use examples to show me how to name an entity? (31116)
What is an occurrence of entity? (31118)
What characteristics do occurrences of entities have? (31119)
Give me some examples of occurrences of entity? (31120)
How do you find the list of properties, identify and name the entities, determine their identifiers, and represent all this information in one operation? (32010)

117

Explanations related

An example of explanation is:

31110: An entity is a set of objects in the conceptual database model that have the same list of **properties** (Q 21180). Some entities represent concrete objects, such as persons and things. Others represent abstract things, such as concepts, categories, events, or situations. Entities may represent objects in their entirety, parts of objects, or different aspects of the same object.

Passive resource

Questions

An example of item is:

31220: What does the entity *<here the name of the entity selected by the user is displayed>* mean?

Explanations related

The explanation related to 31220 is:

31220 The entity *<here the name of the entity selected by the user is displayed>* means that when there is a transaction in regard of *<here the name of the entity selected by the user is displayed>*, the organization records *<here the list of properties is displayed>*. It is true for all *<here the name of the entity selected by the user is displayed>*. Each property of that list has an unique value for each **occurrence of entity** (Q 31118). Is it what you perceived?

Active resource

Rules

The analysis of content did not allow the finding of valid and reliable data in regard of proper identifiers, and of names of properties that can have many values for a same occurrence of entity. Rules associated were not defined. For the purpose of this book, a single rule is presented.

31210: An entity does not have at least one property

Explanations related

The explanation related to 31210 is:

31210: The entity *<here the name of property is displayed>* must have at least one **property** (Q 21160).

Interventions related to unit objective 3

Pedagogical resource

Questions

Table 9.8 presents a part of items related.

Table 9.8

Part of questions related to unit objective 3

What is a relationship? (41110)
What characteristics do relationships have? (41112)
How do you name a relationship? (41114)
Use examples to show me how to name a relationship? (41116)
How do you find the identifier of a relationship? (41128)
Use examples to show me how to find the identifier of a relationship? (41130)
How do you find relationships? (41132)
Use examples to show me how to find the relationships? (41134)
Show me how to find the relationships, using a real situation as an example? (41136)
Can one entity participate in more than one relationship? (41144)
Can two entities be connected by more than one relationship? (41146)
How do you decide whether to represent the reality of the organization by an entity or a relationship? (41160)
How do you find the list of properties, identify and name the entities, determine their identifiers, name the relationships, identify their explicit properties, and represent all this information in one operation? (42010)

Explanations related

The content of the explanation related to item 41128 is:

41128: To find the identifier of an entity, examine its list of **properties** (Q 31110). If a property is an identifier then every different value that it could take will correspond to a different **occurrence of the entity** (Q 31118)

Passive resource

Questions

One of the items displayed in the menu is:

41220: What is the identifier of the relationship *<here the name of the relationship selected by the user is displayed>?*

Explanations related

The explanation related to item 41220 is:

41220: The identifier of the relationship *<here the name of the relationship selected by the user is displayed>* is made of the concatenation of **identifiers of entities** (Q 31122) of its **collection** (Q 41152), being *<here, identifiers are displayed>*. Is it what you observed?

Active resource

Rules

The rule 41312 is formulated as follows:

41312: A relationship has less than two legs

Explanations related

The explanation related to rule 41312 is:

41312: The relationship *<here CODAMA shows a relationship in the model>* does not have at least two **legs** (Q 41138).

Interventions related to unit objective 4

Pedagogical resource

Questions

Table 9.9 presents a part of items related.

Table 9.9

Part of questions related to unit objective 4

What does functional integrity constraint mean? (13110)
Show me how can we know there is a functional integrity constraint?(13120)
How do we represent, using Merise formalism, a functional integrity constraint? (13130)
Use an example to show me what is a functional integrity constraint? (13140)

Explanations related

Explanation 13110 is formulated as follows:

13110: Functional integrity constraints (FIC) are used to take into account the problems of uniqueness for relationships of three or more **dimensions** (Q 41156). Specifically, functional dependencies may appear between certain **entities** (Q 31110) of the **collection** (Q 41152), called the source entities, and another entity, called the target entity. When two entities being in a three-dimensional relationship form a FIC on the third entity in this relationship, the **identifier of the relationship** (Q 41122) is the concatenation of the identifiers of the two initial entities, and not the concatenation of all three identifiers of entities.

Passive resource

Questions

One of the items appearing in the menu is:

13240: What does mean FIC *<here the name of the functional integrity constraint selected by the user is displayed>* ?

Explanations related

The explanation related to item 13240 is formulated as follows:

13240: FIC *<here the name of the functional integrity constraint selected by the user is displayed>* means that in this organization, when they know the values of *<name of one entity of the initial set>* and *<name of another entity of the initial set>*, they also know which value *<name of the target entity>* has. The **identifier of the relationship** (Q 41122) *<name of the relationship>* is the concatenation of **identifiers of entities** (Q 31122) *<names of entities of the initial set>*, being *<identifiers of entities are displayed>*. Is it what you observed?

Active resource

Rules

The rule 41312 is formulated as follows:

41312: Cardinalities used are incompatible with the FIC

Explanations related

In regard of rule 41312, the explanation will be:

41312: FIC *<here the name of the functional integrity constraint is displayed>* is incompatible with the **cardinalities** (Q x, x= 1 (12103 to 12111)).

1.9.2 Identification of explanations presenting examples and statement of examples

Within unit objective 1, four explanations propose examples. Seven from unit objective 2, thirteen from unit objective 3, and two from unit objective 4, propose examples.

In those examples, the organizational context will be presented. Explanation 21180 is used here as example.

. 21180

Presentation of the context in accordance with the following list:

a- Kind of organization.
b- Kind of clients.
c- Kind of transaction.
d- Description of the transaction and of what the organization memorizes in regard of that transaction.
e- Form if available.

The following text will then be displayed:

We have seen a transaction between [...] and a part of its environment. The smallest elements of information that the company wants to store must now be identified.

- For each [...], the company has a [...], [...], [...], etc.

So [...]
[...]
[...]

are properties.

But, [...] is not a property, because [...] are described in the organization by **smallest elements of information** [...].

etc.

Note that [...] will be filled by experts.

1.9.3 List of keywords

Examples of explanations were presented at sections 1.9.1 and 1.9.2. In those explanations, bold characters were used for some words which are followed by a number (an identifier) between parenthesis (Such as **property** (Q 21180)); they are keywords. In the real files, the complete list is presented.

1.9.4 User model description

The description is presented at section 1.9.6.

1.9.5 Fixing errors level allowed

The success level is setted at 80%.

1.9.6 User model functioning

To be able to produce the user model, each error identified by the active resource, and each consultation made by the user are recorded. These data are used to picture the cognitive structure of the user.

Suppose that a user is solving a case. This user elaborated a conceptual database model having 60 properties, 11 entities, 11 identifiers of entities, and 56 properties of entities. There are also 16 relationships, and 4 explicit properties of relationships. There are, at this point, 40 cardinalities. During the elaboration of the conceptual database model, the user called CODAMA; his or her consultations, time between each, and the order of the consultations were recorded. CODAMA produced a user model (cognitive structure) related to the various objectives. Part of the recorded data and of the user model related to unit objective 1 are presented at table 9.10; since the format of the recorded data for unit objectives 2 and 3 is similar, the corresponding tables are not presented here. It can be seen from table 9.10 that CODAMA records, for unit objective 1, all the useful values to be able to make hypothesis and take decisions.

Table 9.10

Part of the recorded data and of the user model related to unit objective 1

U.O.1
Total number of properties: 60
Total consultatic s allowed: 12
Total consultations made: 6

U.O.1						
	T.1			T.2		
	Int	Menu	Keyword	Int	Menu	Keyword
I.1	21110			21150		1
1.2	21120		1	21160 21165	1	1
I.3	21130		1	21170 21175 21177	1	
I.4	21140			21180 21185		
I.5	n.a	n.a	n.a	21190		

Sequence: 21160 menu/ 21130 keyw/ 21120 keyw/ 21150 keyw/ 21170 menu/ 21160 keyw.

Even if cardinalities are part of the unit objective 4 that was presented at section 1.8.6, data recorded in regard of this objective are presented here. In regard of unit objective 4, we mentioned that 40 individual cardinalities had been setted. Table 9.11 presents a part of the data recorded by CODAMA. The related part of the user model is presented at table 9.12.

Table 9.11

Part of the recorded data related to unit objective 4

Total number of individual cardinalities: 40
Total consultations allowed: 8
Total number of pairs of cardinalities: 20
Total consultations allowed: 4
Total number of minimum cardinalities 0: 20
Total consultations allowed: 4
Total number of minimum cardinalities 1: 18
Total consultations allowed: 4
Total number of maximum cardinalities 1: 15
Total consultations allowed: 3
Total number of maximum cardinalities n: 23
Total consultations allowed: 5

		Minimum				
		0	1	$\underline{1}$	c	n
M	1	8	7	x	x	x
a	$\underline{1}$	x	x	2	x	x
x	c	0	0	x	0	x
i	n	12	11	x	0	0

Total number of pairs 0,1: 8
Total consultations allowed: 2
Total number of pairs 0,n: 12
Total consultations allowed: 3
etc.

Table 9.12

Part of the user model related to unit objective 4

Consultations made for individual cardinalities: 2
Consultations made for minimum cardinalities 0: 1
Consultations made for minimum cardinalities 1: 0
Consultations made for maximum cardinalities 1: 1

		Minimum				
		0	1	$\underline{1}$	c	n
M	1	1	0	x	x	x
a	$\underline{1}$	x	x	0	x	x
x	c	0	0	x	0	x
i	n	0	0	x	0	0

Total consultations made for pairs of cardinalities: 0
etc.

1.10 Keywords network elaboration

Links between explanations exist without regard of the hierarchy. The user is allowed to have an access to explanations without using the sequency of objectives determined when the analysis of content was done and used by CODAMA to make hypothesis on the user's knowledge level.

Explanation 41128 (U.O.3/T.2/I.3) will be ⌄ed as example.(Remind that the number 41128 is an identifier; 31128 and 41152 are also identifiers. These numbers are not displayed to the user). Explanation 41128 is formulated as follows (Boulet, 1988c, p.11):

"The identifier of relationship is not setted by the analyst, as it was for the **identifier of entity** (31128). Because the identifier of a relationship is made from identifiers of entities of ᵢts **collection** (41152), it is implicit".

It can be seen that this explanation refers to identifier of entity and collection. Being supposed that the user selects the keyword "identifier of entity", the explanation associated will came from unit 2, i.e., an answer to the question "What is the identifier of entity?". Having had this explanation, the user could either decide to come back to the explanation related to the identifier of relationship, or to select one or many other keywords in the explanation associated to the identifier of entity, and so on.

The extract from the real network presented at figure 9.7 illustrates links between explanations. It can be observed that an explanation may not have any link; explanations 13120 and 41150 are examples. It can be seen at figure 9.7 that an explanation may have only one or many direct predecessors without having any successors; explanation 21130 is an example. In the same way, an explanation may not have any predecessor but, may be predecessor of one or many explanations; explanation 31119 and 41148 are examples. Finally, a same explanation may be predecessor of one or many explanations and have one or many predecessors; explanations 21160 and 31122 are examples.

Only identifiers of explanations are indicated at figure 9.7. Doted arrows and italic characters represent repetition made to avoid a too large number of intersectings.

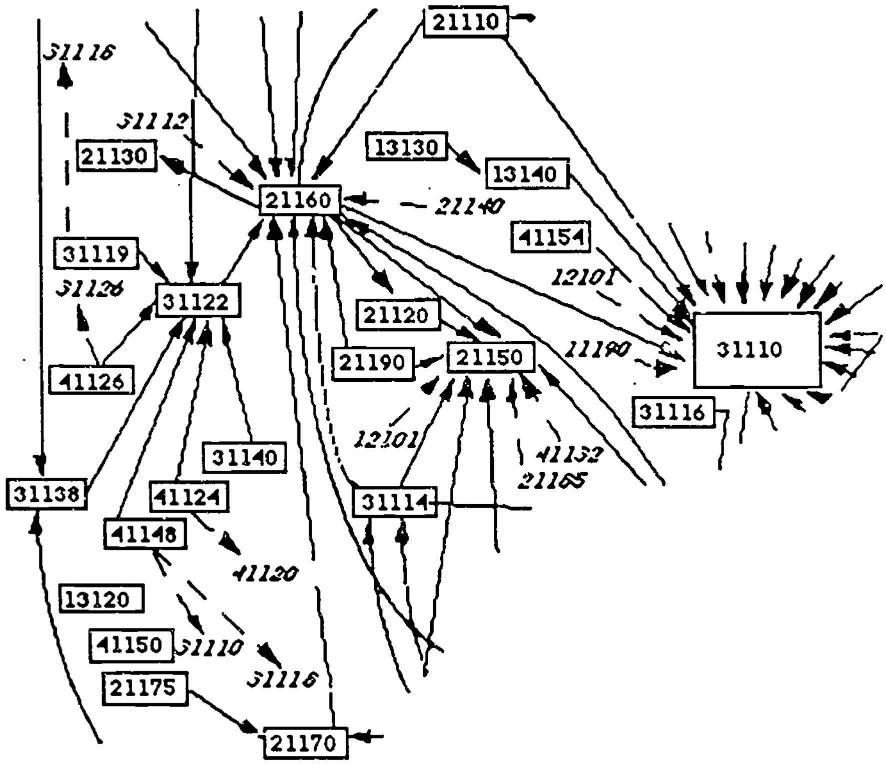


Figure 9.7 Extract from keywords network.

Note that several tables being produced after analyzing the network are presented in the real files. An example is presented at table 9.13. These tables are used when CODAMA analyzes data collected in regard of the user consultations path. Several sets are taken into account when CODAMA makes inferences and hypothesis in regard of the user knowledge level in order to be able to produce the made-to-order training. The detailed description of the works done in regard of formalizing the selection of prerequisites that will be incorporated within the made-to-order training can be found in Boulet, Lavoie, Slobodrian and Tang (1989).

Table 9.13

Pre-calculated table of explarations immediately successor

11110:	31110	31118	41110	41112	41118	41120	41138	41144
11120:	31118	41110						
11130:	31118	41110						
11140:	41120							
11145:	41138							
11150:	41118							
11160:	31110	41118						
11170:	41118							
etc.								

1.11 Hierarchy with intervention/objective links

In the real files, intervention/objective links are presented for each resource. For the purpose of this book, only a part of those related to ‿he pedagogical resource are presented at figure 9.8.

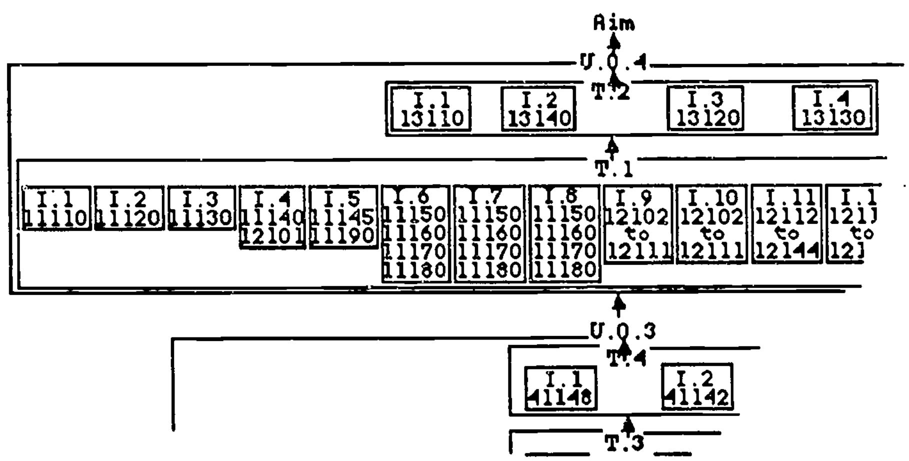


Figure 9.8 Example of intervention/objective links.

2. Use of content analysis

2.1 Users performing task

Ten persons were observed. Real files give details about the method.

2.2 Nature of difficulties

List of questions and rules has been used to indicate the frequency of requests for help. Those frequencies are mentioned in the real files.

2.3 Adjusting content

Several adjustements were done to the content. They are mentioned in the real files.

3. Modelling CODAMA

3.1 Main components identification

Files elaborated according to the content bring to consider that CODAMA has three major parts: an intervention module, a knowledge acquisition module, and a difficulties explainer module (advance organizer).

The purpose of the intervention module is to help an individual using a graphics software when he or she elaborates a conceptual database model. This help can take three forms: active, passive, and pedagogical. CODAMA will answer the user's questions. There are two types of questions: those related to concepts, principles, and rules of conceptual database modelling in general, and those related to the conceptual database model currently developed.

The difficulties explainer module is seen as an extension of the pedagogical resource; it aims at presenting to the user, using an advance organizer, before he or she begins the conception of a conceptual database model, the knowledge related to this field, and at presenting concepts, principles, and rules that give problems to the user during the conception.

The main purpose of the knowledge acquisition module is to allow the storage of real conceptual database model cases. These cases describing real organizations will be used by the intervention module to illustrate theoretical concepts, principles, or rules, allowing the student to have a better understanding of the theoretical explanation.

The next figure (figure 9.9) illustrates that the user has an access to CODAMA via an application software. He or she uses it to elaborate the conceptual database model and to have an access to CODAMA. The intervention module does an analysis of the elements of the conceptual database model produced by the user (how many entities, how many identifiers of entities, how many relationships, etc.). This module also keeps a trace of consultations made by the user in order to be able to determine the proper content of an intervention, or of the made-to-order training. Several explanations present examples; the content of these examples is directly related to those given by domain experts using the knowledge acquisition module.

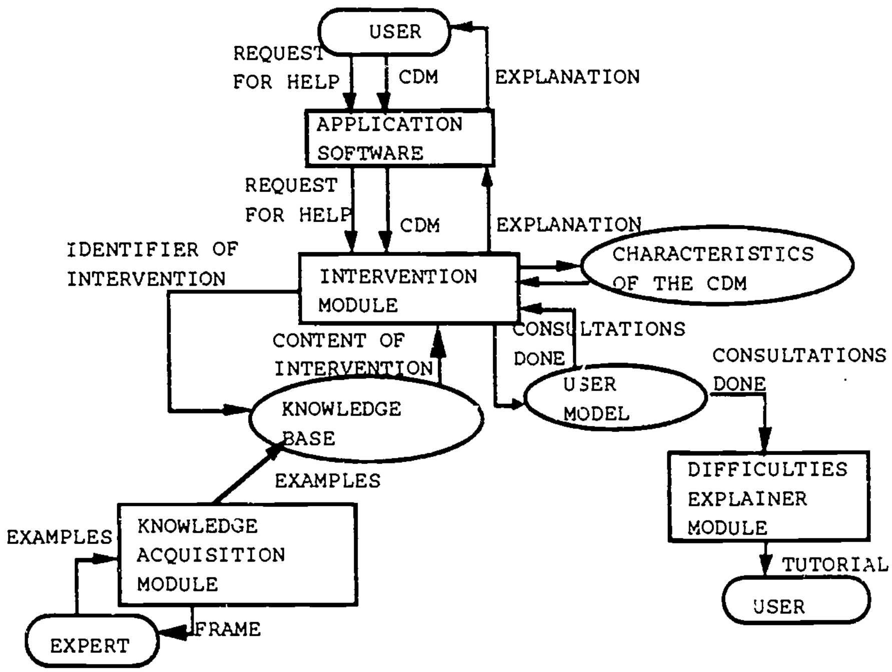


Figure 9.9 Main components of CODAMA

3.2 Processes modelling

It has been decided that structured analysis in accordance with Gane et Sarson (1979) would be done. In the real files, all levels of models are presented with the detailed documents associated to this kind of modelling. Taking into account their wide-ranging, the fact that this technique is well documented in the specialized literature, and that they are not needed to reach the goal of this book, these diagrams are not presented here[4].

3.3 Data modelling

It has been decided that an entity-relationship diagram would be elaborated (Chen, 1976). In the real files, the

---

[4] Note that the data flow diagram of MUSIC will be presented in the next chapter; it is far less complex.

diagram is presented with the detailed documents associated to this kind of modelling. Taking into account the fact that the diagram is very complicated, that this technique is well documented in the specialized literature, and that it is not needed to reach the goal of this book, this diagram is not presented here[5].

4. Additional informations requested by analysts

During the modelling process of CODAMA's data and treatments, analysts requested for a description of the functionin~ of the difficulties explainer module. The following informations were obtained:

In accordance with Ausubel (1960), the advance organizer that will be used either before starting the process of elaborating a conceptual database model, or when no data on the user knowledge level is available, is produced in regard of the hierarchy presented at figure 9.8.

First, CODAMA displays a complete conceptual database model obtained from experts. This model being displayed, the difficulties explainer module simulates, according to the order of prerequisites, the elaboration of this model. In conformity with principles of design of advance organizers (Guéladé, 1987), the length of explanations will be approximately a quarter of those presented by the other part of the pedagogical resource. Let's illustrate this last point.

The first explanation relates to U.O.1/T.1/I.1; the content of the explanation associated, being 21110, is summarized. The following text is superimposed to the diagram:

---

[5] Note that the entity-relationship diagram of MUSIC will be presented; it is far less complex.

"A conceptual database model is a map of an organization's memory".

This explanation is followed by a summary of the one related to U.O.1/T.1/I.2 (being 21120):

"The conceptual database modelling springs from the need to manage information in organizations. Such a model separates information from the processing it undergoes".

In accordance with Ausubel (1960), when the difficulties explainer module intervenes after a user has begun to elaborate a model, it takes into account data related to the user cognitive structure. It uses data collected during the process. A simplified example is presented in the following paragraphs.

Let's suppose that a user referred in many occasions to interventions related to U.O.1/T.2/I.4, U.O.2/T.3/I.2 and U.O.3/T.3/I.2; this user did not use keywords. CODAMA will infer that this user neglects interventions regarding the foundation of the process and looks only to those regarding conception. It will fit its interventions according to this lack of knowledge. It could produce the following advice before displaying the advance organizer:

"During the elaboration of the conceptual database model named **A**, you seek advice 15 times from items related to conception. Your model has 60 properties; you asked 11 times how to find properties. 11 entities can be found in your model; you ask 2 times how to find entities. You ask 2 times how to find relationships; your model has 16 relationships.

I therefore concluded that you have difficulties with fundamental concepts related to database conceptual modelling".

Following this advice, the advance organizer will be elaborated. CODAMA will begin the simulation of the conception of a conceptual database model with the highest element of the hierarchy that was consulted, being in the present example U.O.3./T.3. It will explain the reasons behind the decision to represent such an element in such a way. It will after go down the hierarchy; finally, it will go back all the hierarchy.

CHAPTER TEN

CASE STUDY 2

MUSICAL COMPOSITION ADVISOR SYSTEM: FILES RELATED TO THE DESCRIPTION OF THE CONTENT AND OF THE ARCHITECTURE

1. Analysis of content

1.1 Prerequisites identification

Fourty one prerequisites have been identified. Table 10.1 presents the tentative list of prerequisites, and their links with the secondary level music program (Ministère de l'Éducation, 1981b, 1983).

1.2 Consultation of specialized books

More than thirty books have been consulted. In the real files, the list is presented.

1.3 Synthesis in view of formulating the tentative content of explanations

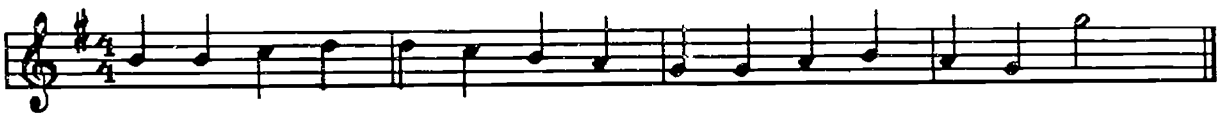
We solely present here two examples of explanations; one has no musical example while the other has. References to other prerequisites are indicated by bold characters. In the real life, this identification and the standardization of the vocabulary are done at the next phase (1.4).

Example 1: ACCIDENTAL

It is a symbol aiming at elevating or reducing of a **semitone** the note it is in front of. The three main accidentals are **flat (#)**, **sharp** (♭), and **natural** (♮).

Example 2: RANGE

It is the distance between the lowest and the highest note of a melody.



Here the lowest note is G, second line; the highest note is high G. The range of the melody is an **octave**.

Table 10.1

Tentative list of prerequisites and links with music program

Prerequisite	Program module	Nb objective	Page
Accidental	Musical language	8.3.1	35
Anacrusis	Creation	9.2.1.1	'8
Answer	Creation	9.2.1	48
Bar line	-----------------	------	--
Bar	Musical language	7.3.1.4	26
Beat	Musical language	7.3.1 3	26
Cadence	Creation	9.2.1	48
	Musical language	9.3.7	50

1.4 Links between prerequisites

Figure 10.1 illustrates an extract from the network. It relates to the question-answer musical composition process.

1.5 Analysis of students' compositions

Ten students did musical composition activities on paper. Their compositions were then analyzed to detect weaknesses. Three music teachers and two composers did comments on them. In the real files, those results are detailed.

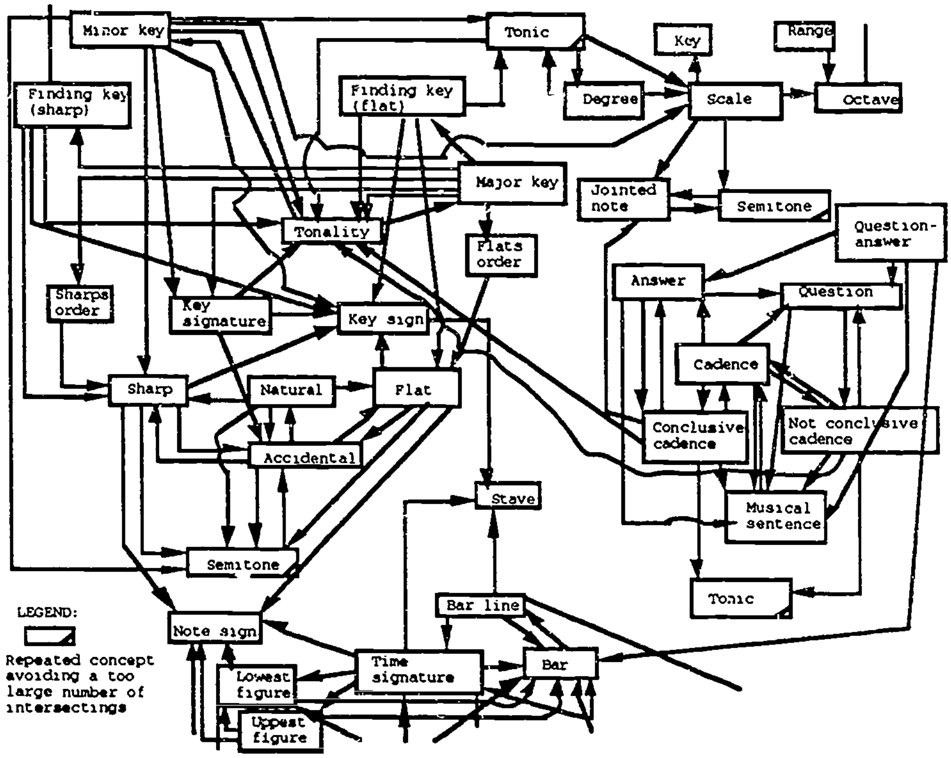


Figure 10.1 Extract from prerequisites network.

1.6 Consultation of specialized books related to composition process

More than fifteen books have been analyzed. In the real files, the list is presented.

1.7 Synthesis to make out a list of determinant elements and identify comments related

1.7.1 Determinant elements

In regard of the question-answer process, ten elements have been identified. Six are considered as being mandatory; they are used to detect weaknesses. The four others only aim at doing comments on the composition. Those determinant elements are describe' in the following paragraphs.

First, it has been found that a balanced melody consisted of two complementary parts with same length (same number of bars) and same number of beats (Ad Lib, 1987; Aubert, 1988; Fournier, Milot, Richard, Béchard, and De Melo, 1986; Warburton, 1960; White, 1869). Consequently, the first determinant element aims at allowing MUSIC to verify if the number of bars of the answer is equal to the one of the question.

The second element aims at allowing MUSIC to verify if the number of beats by bar of the answer is in accordance with the one of the question. Time signature will be used to do this verification.

The third element relates to the second one; it will be considered only if there is an anacrusis at the beginning of the question. The sum of beats of the last measure and of the anacrusis must be in accordance with time signature.

The fourth element relates to the last note of the answer; it must be the tonic. According to Crighton (1980): "Statement: A melody ranging in length from a phrase member to complete a musical idea. If in binary relationship it is dependant for completion upon a second statement called response; ... Binary: A two-part form. Melodically, the parts may be either parallel (AA') or contrasting (AB). There will be movement towards the dominant, the first part ending on dominant harmony or in the dominant key (relative major if the original key is minor). The second part returns to the tonic" (p.37).

Because it concerns the note coming before the tonic, the fifth element has a link with the fourth. The note coming before the tonic must be a jointed one; the two last notes of the answer make up a perfect cadence. That means the second or seventh degree of the scale associated must be the next to last note of the answer: "... the tones before the last must be either the 2nd or the 7th scale-step, so that the last progression is either up or down the scale into the final keynote. Such an ending is called the Perfect Cadence" (Goetschius, 1920, p.4); "the approach to the final note in the melody is 7-8 or 2-1" (Jones, 1963, p.16).

The sixth element relates to the concept of strong beat: "The last note of every phrase must occur on a strong beat " (Warburton, 1960, p.2).

The seventh element relates to notes used on the strong beats of the question; some of those can be found in the student's musical answer. As a musical answer is a kind of rehearsal of a musical question (Schoenberg, 1987; Kraft, 1976), significant notes of the question are used to make up the answer. These significant notes are the ones that are used on the strong beats of the question. Although student can modify the melody of the answer, almost invariably, notes that were used on strong beats of the question will be found on strong beats of the answer (White, 1869).

The eighth element relates to the kind of relation established between the question and the answer: "Response: The second part of a statement-response. The response, which may be a phrase member, phrase, sentence, of one of the divisions of a binary form, completes the musical thought or idea expressed in the statement and may be further identified as being in parallel, semi-parallel, or constrasting relationship with the statement" (Crighton, 1980, p.37). It aims at allowing MUSIC to comment the kind of the answer. The eighth determinant element will allow MUSIC to check how the student uses the first half of the question in his or her answer. For example, a parallel answer can use both melody and rythm found in the first half of the question; another type of parallel response can use the same rythm than the one used in the first half of the question, but modify the melody; another parallel response can use the same melody, but modify the rythm.

The ninth element relates to the range of the melody; it must be in accordance with the one stated in the parameter associated to the statement of the activity. If not, MUSIC interprets it as being a failure to give attention. Therefore, it merely informs the student.

The last element refers to notes used in the answer. White (1869) mentions that the two parts of a musical sentence must be defined by the same tonality or mode. MUSIC will check if notes used in the answer are in accordance with the key signature.

Table 10.2 presents the list of determinant elements that must be abided. A letter is associated to each of them. Table 10.3 presents the list of elements aiming at merely doing comment on compositions; a letter is also associated to them.

Table 10.2

Tentative list of mandatory determinant elements

A	Number of bars of the question = Number of bars of the answer
B	Number of beats by bars in accordance with time signature
C	Anacrusis + Number of beats of the last measure in accordance with time signature
D	Last note of the answer = tonic
E	Next to last note of the answer = 2nd ou 7th degree of corresponding scale
F	Last note of the answer on a strong beat (1st beat $\frac{2}{4}$ or $\frac{3}{4}$; 1st and 3rd beats $\frac{4}{4}$)

Table 10.3

Tentative list of determinant elements allowing comments

G	Notes on strong beat question = Notes on strong beat answer
H	Relationship between the question and the answer
I	Range of the melody in accordance with parameter
J	To what extent notes of the answer are linked to the corresponding scale

1.7.2 Comments related

All combinations of determinant elements with the comment related are presented in the real files. How MUSIC does the analysis is also fully explained. We solely note here that after having attributed a weight to each element or group, a certain grouping and ordaining of mandatory determinant elements were done.

In regard of the analysis of composition, the mandatory determinant elements are checked at first. A first group having the highest weight is considered; if they are not properly used in the student's composition, MUSIC does a relevant comment and ends the analysis. A relevant comment relates to a certain combination that a group of determinant elements has. The main concern is to focus the attention of the student on his or her weaknesses; if the comment is too long, as it would be if it took into account all the determinant elements, it will puzzle the student. When the result of the analysis shows that this first group of elements is properly used, a second group of mandatory determinant elements is verified; again, if the result of the analysis shows that some of them or all are missing, a relevant comment is done, and MUSIC ends the analysis. When the list of mandatory determinant elements is empty, MUSIC begins the analysis aiming at only commenting certain characteristics of the melody. When MUSIC does comments on its own composition, it only considers these last determinant elements.

The following is an example of a comment being displayed when an A=0, but other variables have the value 1: "After having analyzed your composition, I found that your **answer** does not have the same number of **bars** than the **question**. Your answer, as the question, should have ___ bars" (Bold characters correspond to keywords). Student is allowed to select one or many keywords, to ask MUSIC to compose and do comments on its composition, or to merely begin a new exercise.

1.8 Sequencing the activities in accordance with the spiral curriculum

In order to allow the music teacher to set out his or her own sequence of activities in accordance with the spiral curriculum, MUSIC must be able to record data in regard of note and rest signs, keys, time unit, etc., allowed. Options of the software toolbox will be adjusted to fit these data.

2. Modelling MUSIC

Structured analysis was used to elaborate models of the advisor. Taking into account results of the previous steps, analysts divided MUSIC into five modules: a module aiming at presenting explanations, a module aiming at presenting comments, a module aiming at composing, a module aiming at analyzing compositions, and a module aiming at collecting statements and characteristics of activities. The data flow diagram (global) elaborated, using Gane and Sarson's approach (1979), is presented at figure 10.2.

In the real files, the content of each data store is detailed. As example, the data store "Student's composition" is described as follows:

Number of bars question
Time signature
Tonic (derived by key signature)
Lowest note (question + answer)
Highest note (question + answer)
Range of the melody (question + answer)
Value of the first note

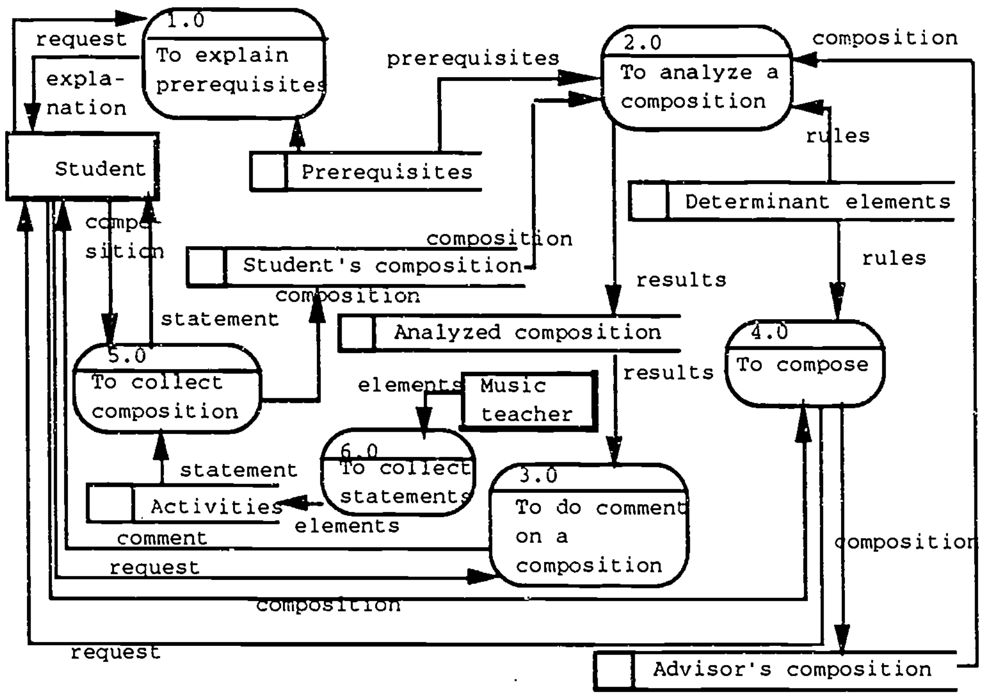


Figure 10.2 DFD (global) of MUSIC.

In regard of data modelling, an entity-relationship diagram was elaborated (Merise formalism). It is presented at figure 10.3.

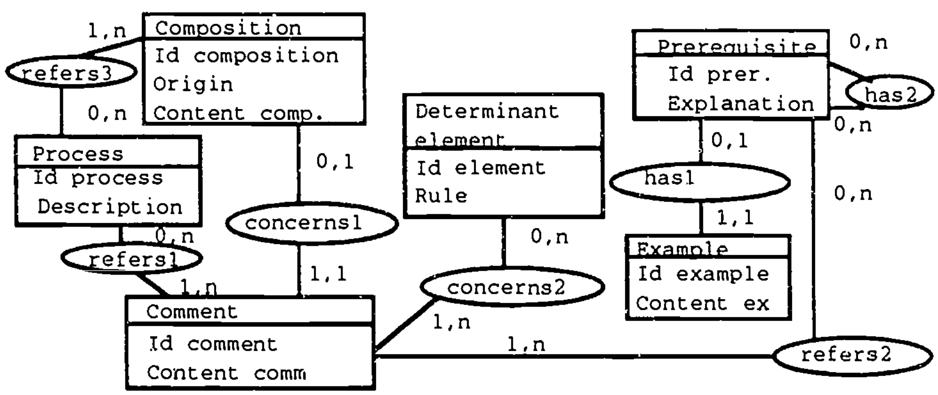


Figure 10.3 Entity-relationship diagram (First draft).

Note that these two models were detailed as works go along; they were also well-documented in a dictionary.

CHAPTER ELEVEN

STEP 4. DESCRIPTION OF THE PHYSICAL IMPLEMENTATION

Works in regard of the physical implementation begin after the logical blueprints of the advisor have been approved. Works to be done at this step involve hardware (machines), software (computer programs, procedures, forms), and peopleware (personnel). While implementing the advisor, problems that had not been anticipated during the three previous steps often appear. Solutions to these problems usually require modification of the original design.

Getting the advisor to the operational stage

During step three, it is possible that data flow diagrams and entity-relationship diagrams have been produced. Whatever means were selected to do the analysis, analysts did logical modelling of the advisor. According to Gane and Sarson's approach (1979), analysts have to convert the logical model of the advisor into a physical model. First, analysts partition this logical model of process and data into procedure units, i.e., chunks of automated and manual procedures that can be executed and developed as units. To do this, they consider each input and output, and ask the following questions for each one:

- When does it happen?

- How large an area of DFD is involved in handling and producing it?

- Can that area be implemented as a single unit? If not, why not?

Logical model of process and data being partitioned into procedure units, analysts have to specify the details of each procedure unit that will be required to implement the advisor. A procedure-unit specification may involve:

1. an extract from the advisor DFD showing where this procedure units fit into the rest of the advisor;

2. details of the tables accessed by the procedure unit;

3. layouts for any screens and outputs involved in the procedure unit (can be prototyped);

4. details of the logic and procedures to be implemented, written in some unambiguous form.

With the nature of the procedure unit defined, analysts can decide whether it should be prototyped or implemented directly in the target language. Case study one is an example of advisor having been implemented directly in C; case study two is an example of advisor having been prototyped before programming the whole advisor.

Getting the advisor to the operational stage involves separate and distinct tasks that must be performed in sequential order:

1- Program the advisor;

2- Test the programs;

3- Implement the advisor.

Program the advisor is the point at which application programs are written in order to perform whatever help functions are being computerized. If some of the programs or program modules are being purchased, this task may be shortened. The programmers must study the design details of the advisor in order to understand how it should operate. Analysts who performed the study should be available for advice. Programmers and analysts can decide together which programming language(s) to use.

Test the programs involves the testing of the programs, a full advisor's test, and the documentation of the programs. Methods proposed in the literature are used.

Implement the advisor is the point at which the programs are run, the different files are interfaced, and the human-interface is setted.

Tasks of this step are summarized at table 11.1.

Table 11.1

Summary of the fourth step

1.	To convert logical models into physical models.
1.x	*Steps of the method selected.*
2.	To validate the content of interventions.
3.	To program the advisor.
4.	To test the advisor.
4.x	*Steps of the method selected.*
5.	To implement the advisor.

CHAPTER TWELVE

CASE STUDY 1

CONCEPTUAL DATABASE MODELLING ADVISOR: FILES RELATED TO THE PHYSICAL IMPLEMENTATION

Implementation

CODAMA has been implemented on an IBM™ PS2 computer. It has been interfaced to WINDOWS™ graphics software. The C language was used for programming the various modules. To allow the programming of the knowledge, a subset of C, called Hyperfac language, was developed.

It can be seen at figure 12.1 that an interface package is used as a data link between the Hyperfac program and the application program. Link A is an action data path through which the interface package may trap user data and actions. Link B is the Hyperfac programs' information request line. Link C is the Hyperfac programs' information acquisition line. Note that an interface package may be written for any application software, thus allowing a Hyperfac program to be linked to any application.

Figure 12.1 Linking Hyperfac program with application program.

The various components of a Hyperfac program are: User variable declarations, User function definitions, Facet declarations, Index declarations, Program source blocks.

The user variables are limited to the C language simple integer types pointers and arrays of these. Complex types (structures and unions) will be successfully parsed and may be referred to in the function definitions but not in the program source block. The user function definitions refers to functions which make up the interface package module. The facet declarations is used to state all facets used by the program. The index declarations refers to suffix into the facets. The program source blocks are independently executable blocks of code containing text and control structures.

Figure 12.2 illustrates that domain experts (at the right top of the figure), being regular members of the team project, provide the content of explanations, objectives, questions, rules, etc., to a programmer; it can be ASCII files produced by a word processing software. In regard of the pedagogical resource, it is the theoretical part of the explanations that is provided, not the variable part aiming at presenting examples. Programmers use either C or Hyperfac to code interventions. As Hyperfac is a subset of C, it is easier to learn. So, an expert can easily use it (and did use it in our case) to code what is identified in files issued from the previous step as *explanations related*. In regard of menu items, objectives, and rules, they must be coded in C programming language. In addition, all explanations having variables, after being coded in Hyperfac, must then be adjusted in accordance with C programming language. In fact, Hyperfac language allows solely the coding of keywords, and links associated. At the left top of the figure 12.2, it can be observed that domain experts, (may not be regular members of the team), being appointed to the production of examples, use the knowledge acquisition module. An automatic code generator produces Hyperfac programs associated. The various programs are compiled by the appropriate compiler, either the Hyperfac or C compiler. As illustrated at figure 12.2, compiled programs are after jointed to produce a module for CODAMA. The program is then executed using a virtual machine.

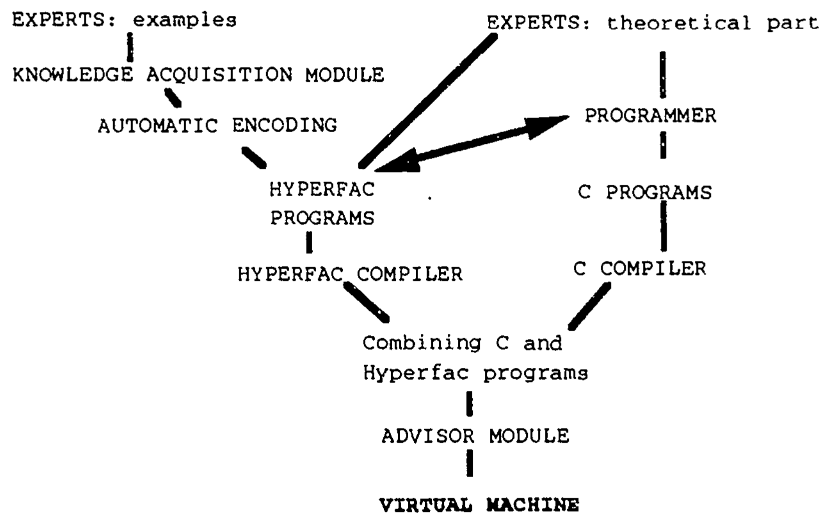


Figure 12.2 Program structure.

Figure 12.3 illustrates the components of the application. It refers to a hypertext compiler which generates C code as the target language along with the usual virtual code. The Hyperfac compiler can produce code for any target language.

In regard of the internal program organization, application actions, data, and requests, are routed through the interface package (figure 12.4). Function calls made by the virtual machine correspond to calls to the interface package. It allows the use of different target languages. The Btree index is used to accelerate searches in the Hyperfac image file. The Hyperfac image file contains code and text. The code is loaded into virtual memory by the interface package, and then executed by the virtual machine. The facet headers contain information such as: the facet disk address, the number of dimensions, and the dimensions size. They are used by the interface package to access the different elements of each facet. A LRU cache is used to minimize unnecessary swapping.

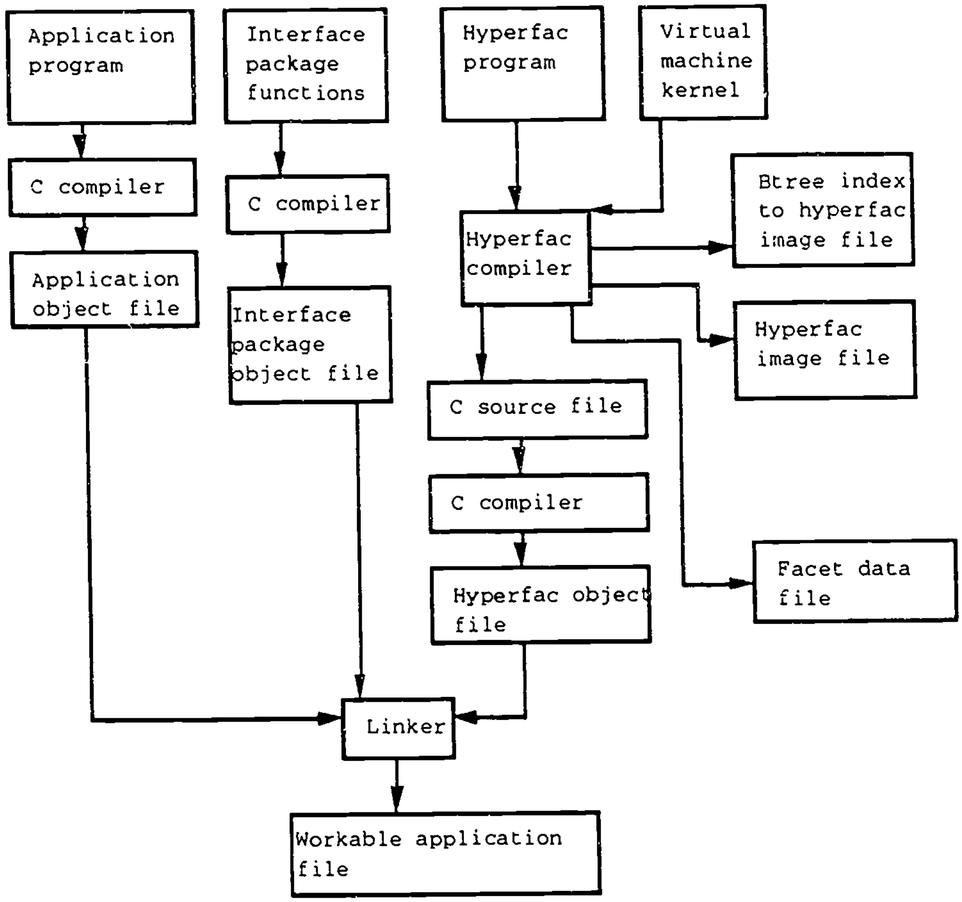


Figure 12.3 Components of the application.

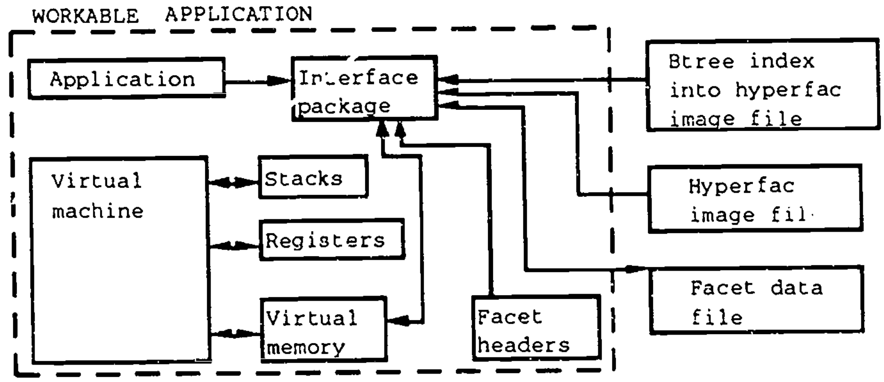


Figure 12.4 Internal program organization.

BEST COPY AVAILABLE

The virtual machine is divided into four parts: the central processing unit (CPU), the memory, the registers (R0 = the flag register), and the stacks (figure 12.5). The number of bytes of code memory, bytes of data memory, registers, and stacks are tailored by the compiler to suit the application. For example, the number of registers depends on the complexity of expressions; the size of the data segment memory depends on the number of variables declared. Function calls push both the flags R0 and the return address on stack 0. A full set of instructions are interpreted by the CPU allowing very complex expressions to be handled by the compiler. All support functions (minus size definitions) are contained in the virtual machine kernel[1].

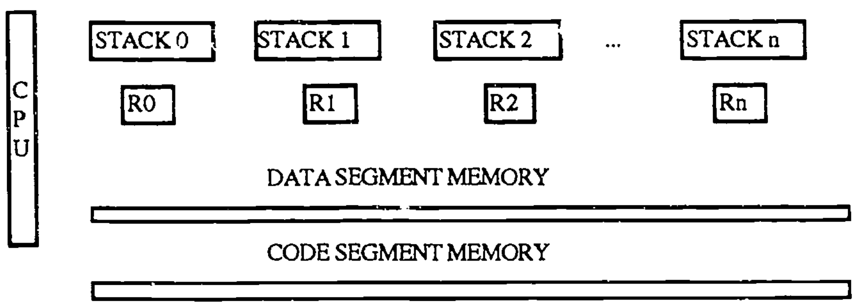


Figure 12.5 The virtual machine.

Functioning of CODAMA

Figure 12.6 shows the window of the graphics software being used. It can be seen that the set up of CODAMA adds an option within the menu bar: the ADVISOR option. In this figure, it can be seen that a user has begun to elaborate a conceptual database model. Entities ADDRESS, COURSE, PROFESSOR, PROGRAM, and STUDENT have been created. Relations ENROLLS, GIVES, and

[1] In the real files, instructions of the virtual machine are fully detailed. In the same way, all the technical aspects are described.

HAS have also been created. There are properties that have not been yet associated to an entity or a relationship. Some cardinalities have been setted.

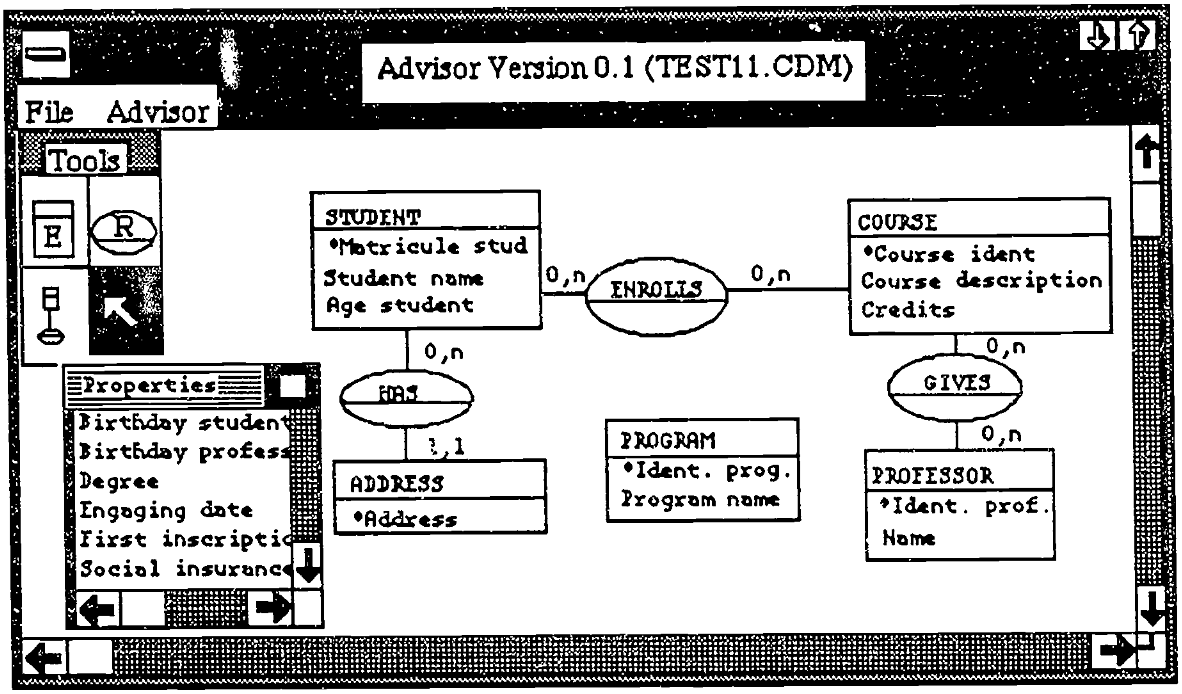


Figure 12.6 The graphics application software.

Figure 12.7 shows the main menu displayed to the user, after he or she selects the ADVISOR option. It allows the user to select the type of resource.

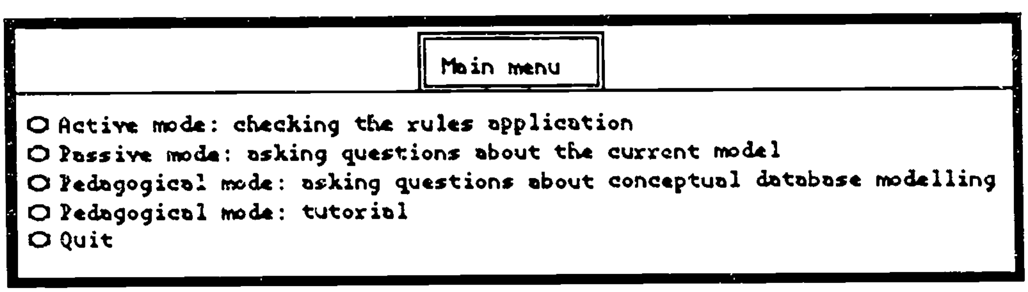


Figure 12.7 Main menu.

If the user selects the passive mode, the window illustrated at figure 12.8 is displayed. Note that items of this menu are closely linked to unit objectives having been identified.

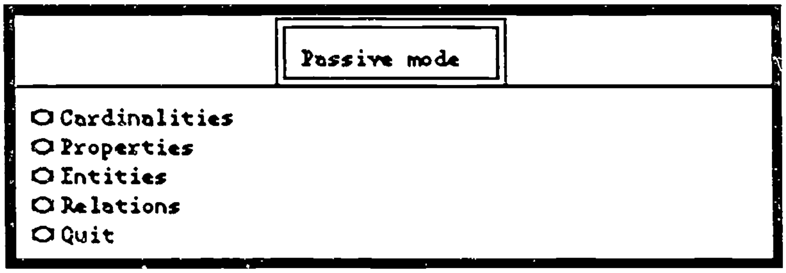


Figure 12.8 Selecting the topic (passive mode).

Suppose that the user selected "Cardinalities" (figure 12.8); that means that he or she wants to have some explanations related to cardinalities. The box "click on a relation" illustrated at figure 12.9 allows this user to point out for which cardinalities he or she wants an explanation.

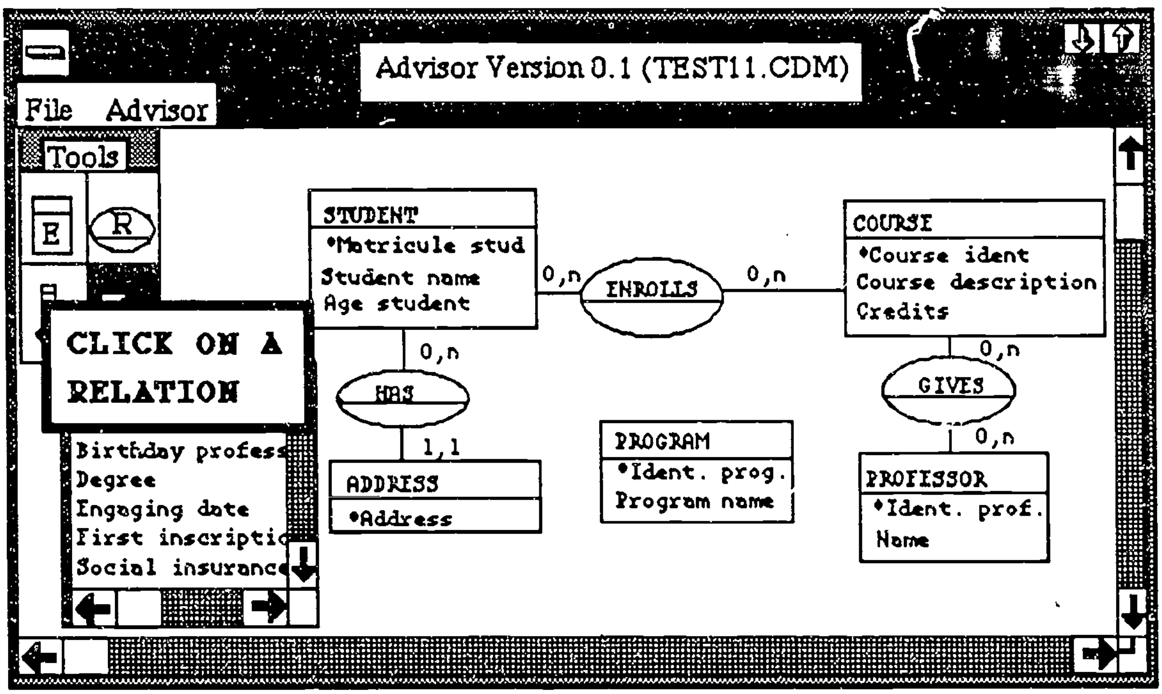


Figure 12.9 Selecting which cardinalities will be explained.

Suppose that the user clicks on the relationship ENROLLS (figure 12.9). The menu presented at figure 12.10 will be displayed. Note that items of this menu are in accordance with the analysis of content.

BEST COPY AVAILABLE

Passive mode: CARDINALITIES

- ○ Explain couple 0,n-0,n
- ○ Explain pair 0,n of entity STUDENT
- ○ Explain pair 0,n of entity COURSE
- ○ Explain minimum cardinality 0 of entity STUDENT
- ○ Explain maximum cardinality n of entity STUDENT
- ○ Explain minimum cardinality 0 of entity COURSE
- ○ Explain maximum cardinality n of entity COURSE
- ○ Quit

Figure 12.10 Selecting the intervention.

If the user clicks on the option "Explain couple 0,n-0,n", a part of the explanation that will be superimposed on the conceptual database model is:

"The two **entities** STUDENT and COURSE can exist even though the **relationship** ENROLLS may not exist. When the relationship ENROLLS does exist, the two entities STUDENT and COURSE must exist at the same time with respect to the relationship FOLLOW, and that there is no **functional dependency** between these two entities. [...]".[2]

Functioning of the knowledge acquisition module

The knowledge acquisition module is the part of CODAMA that allows the recording of real examples of conceptual database models according to a specific domain. In order to acquire those examples, experts in conceptual database modelling firstly use a graphics software (Windows™) to draw the model that will be used as example. Once an expert has completed the drawing of the model, CODAMA's knowledge acquisition module is activated. As illustrated at figure 12.11, it asks the expert to describe the organizational context.

---

[2] In the real files, the functioning of each other mode is described and documented. As their functioning is similar to the passive mode, they will not be presented here. The functioning of the knowledge acquisition module being different, it will be summarized.

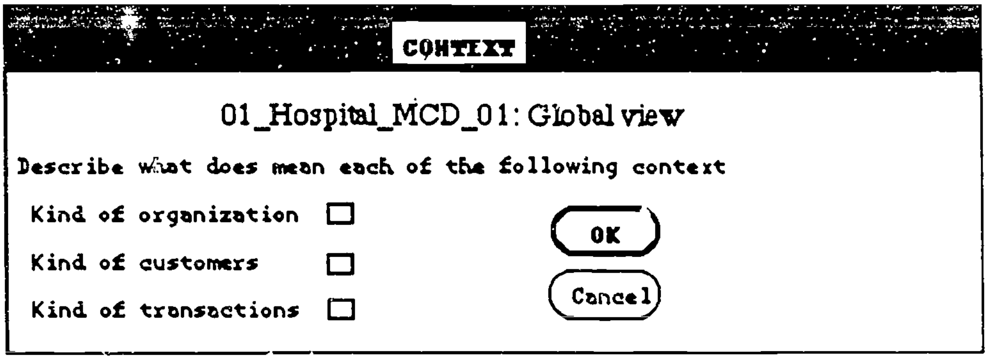


Figure 12.11 Collecting information about the organizational context.

First, the kind of organization is described. Figure 12.12 presents the window used to collect the data related.

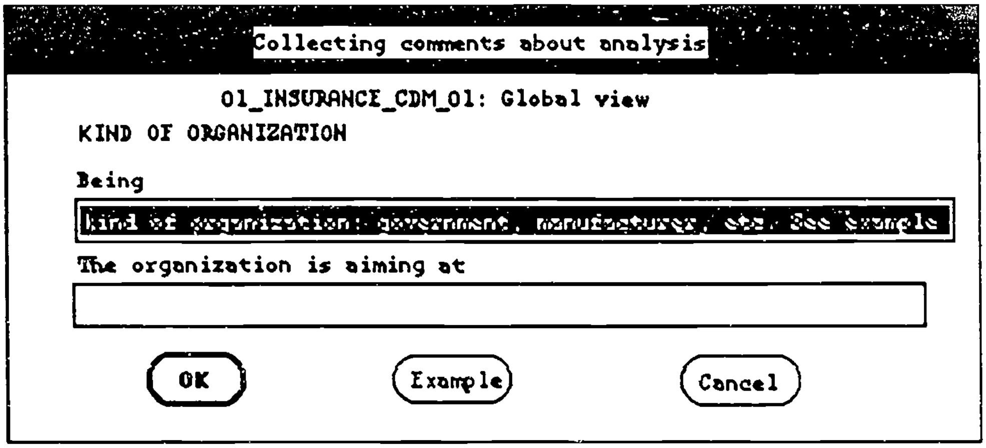


Figure 12.12 Collecting information about the kind of organization.

It can be seen at figure 12.12 that the expert can see an example. Figure 12.13 illustrated this option.

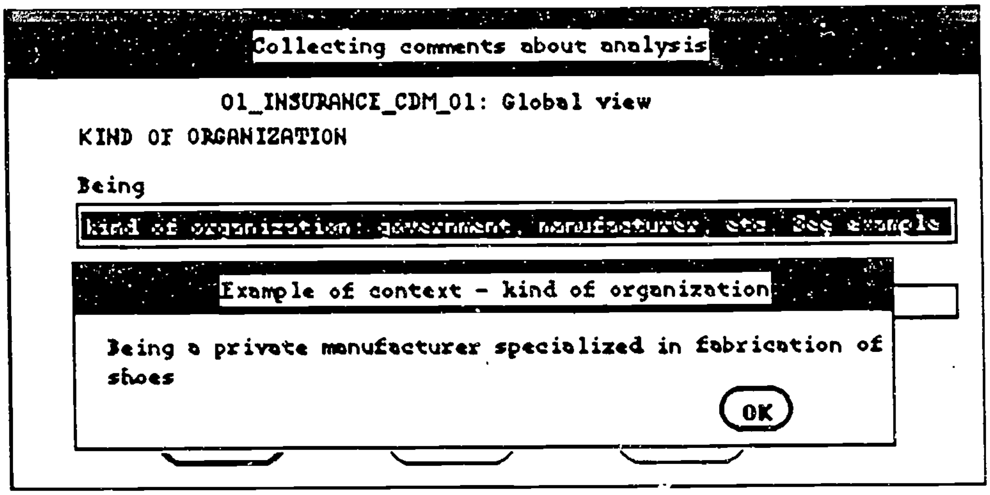


Figure 12.13 Example related to the kind of organization.

Following this description, the kind of customers the organization has, and the kind of transactions will be described. Windows used to collect these data being similar to those used for the kind of organization, they will not be presented here.

The organization being described, a specific transaction has to be described in regard of each unit objective. Figure 12.14 shows an example of information collected in regard of unit objective 4 (Cardinalities). Note that an example is available.

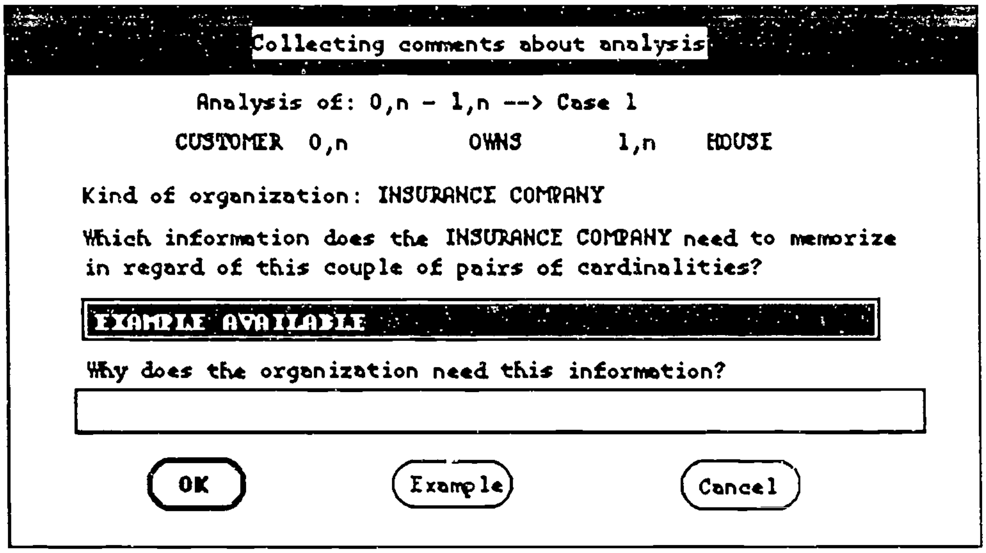


Figure 12.14 Collecting information about cardinalities.

CHAPTER THIRTEEN

CASE STUDY 2

MUSICAL COMPOSITION ADVISOR SYSTEM: FILES RELATED TO THE PHYSICAL IMPLEMENTATION

General remarks

Before programming the whole advisor, it has been decided to develop several prototypes. Reasons are expounded in the following paragraphs.

Domain experts elaborated at the previous step, a first version of explanations. As it is especially hard to make experts understand that the access to a same explanation is through several different ones, so it must be expressed in order to have a meaning wherever a student comes from, and as the screen is limitated in regard of the space available, Hypercard was used to quickly develop a first prototype. The prototype being developed, content and form of explanations were modified to a considerable extent. At first, domain experts setted out a quite bookish content. For example, concept of beat was defined in the following way: It is each equal division of a **bar**. However, using the prototype, they realized that this definition was not very helpful. In the same way, the prototype being used by a sample of future students, this explanation and some others were not really understood; to assess the understanding, it was asked to each student to state in his or her own words what was the meaning of the explanation displayed. The way used to explain the concept of beat was modified; it became: Beat indicates duration, space bounded by two **pulsations**. You can refer to examples of **2 beats bounded by 3 pulsations**, of **3 beats bounded by 4 pulsations**, and of **4 beats bounded by 5 pulsations**. Network was adjusted; that means that concepts of pulsation, 2 beats bounded by 3 pulsations, 3 beats bounded by 4 pulsations, and 4 beats bounded by 5 pulsations, have been added. The prototype having been tried by subject matter experts and future students, several other adjustments have been done. Eighteen prerequisites were added.

A second prototype regards the module allowing the analysis of compositions; it aims at allowing MUSIC to do comments on compositions. Domain experts formulated rules at step three (determinant elements); those rules were implemented using BNR Prolog. That allowed experts to realize that many rules were missing; those rules being added, associated prerequisites were defined and added in the network. Twelve rules were added with all the linked comments; six prerequisites were added in the network (and to the first prototype).

A third prototype regards the creation of compositions by MUSIC. BNR Prolog was used to develop it. Note that the prerequisites and rules discovered when the previous prototypes were used have been added. But, domain experts realized that seven other rules were missing. In the same way, they identified two new prerequisites. The seven rules and the two prerequisites with the explanations related were added to the first and second prototypes.

To summarize, at the end of the prototyping phasis, there were sixty seven prerequisites, and twenty nine rules. Before programming the whole, the final version of each prototype was used in a pilot study aiming at collecting information about the reaction of the target students. Adjustments were made to the design.

The prototyping phasis being completed, the three prototypes have been integrated into a whole; MUSIC has been programmed in Pascal programming language using an IBM™ plat.orm, and interfaced to a musical writing software. Compiled code and mechanisms of constraints put in the inference process avoiding a combinatory explosion when MUSIC do researches have been setted out. In regard of the management of memory space, processes of elimination of irrelevant data, and of recording, have been installed. MUSIC can be used on the following IBM™ computers (or 100% computers-compatible): IBM PC XT, IBM PC AT, IBM PS2, model 30, 50, 60 or 80, with 640K of RAM memory. The Disk Operating System (DOS) release 3.3 or more must be used. Graphic cards that may be used are: PC3270, ATT400, HERCULE, CGA, MCGA, EGA et VGA. It is easier to operate MUSIC with a Microsoft™ mouse.

Modelling software architecture

In this section, MUSIC's basic components are presented; links and communication between these components are also described. In the first part, the States Machine Diagram (SMD) are presented. They are used to describe components and links between them. Communication links between components are illustrated through the description of MUSIC's main menus.

MUSIC blueprints

Figure 13.1 diagram illustrates processes of musical composition that can be used. When the execution of MUSIC begins, MUSIC detects and initializes graphic mode parameters in regard of the graphic card used. The main menu is then displayed. There are several options available: to compose according to question-answer process, to compose according to augmentation, diminution, and repetition processes, to create composition activities, and to end a work session. In any case, the user returns to the main menu as long as he or she does not select the option End of the work session.

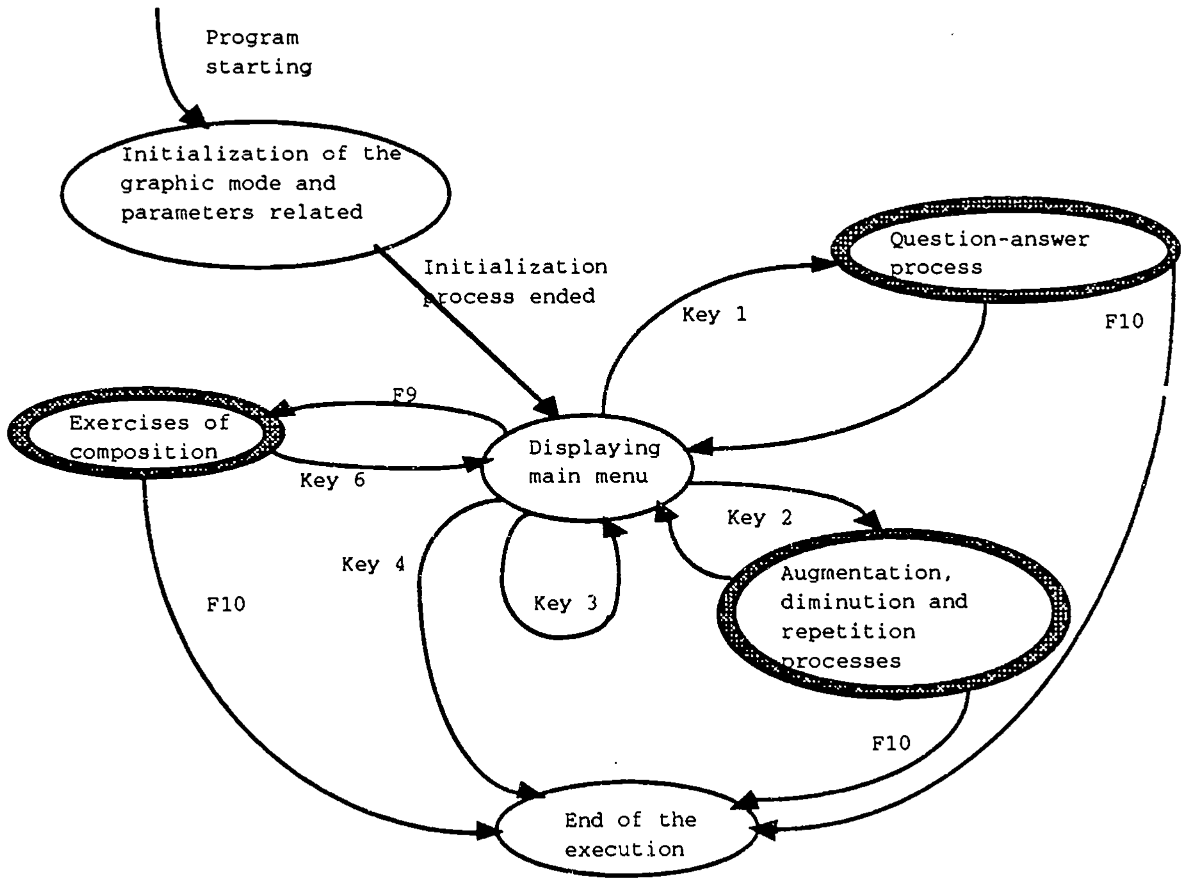


Figure 13.1 Program starting up.

Figure 13.2 regards the execution related to the question-answer musical composition process. First, the statement of the exercise is displayed as long as the student does not press on the return key. The musical question is then displayed. The student can begin to compose his or her response. To start MUSIC's analysis of the composition, the student presses on F8 key. MUSIC will then check out if there are other exercises in the file. If there are some, the same process starts again; if not, the main menu is displayed. To quit, the student uses the F10 key.

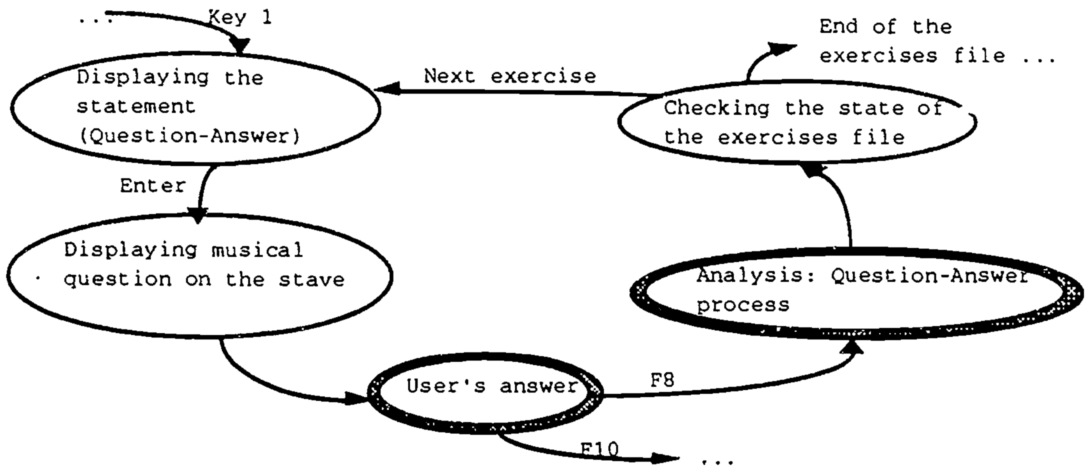


Figure 13.2 Question-answer process.

Many options allow the student to compose his or her response. They are illustrated at figure 13.3: "1" corresponds to moving backward the highest stave, "2" to moving forward the highest stave, "3", to moving backward the lowest stave, "4", to moving forward the lowest stave, RETURN, to adding a musical symbol on the stave, F8, to analyzing and doing comments on a composition, F10 corresponds to ending a work session, ALT-O, to the opening of the menu OPTIONS PIANO, ALT-F, to the opening of the menu FUNCTIONS. All the options proposed within each menu can also be activated by pressing on the keyboard a function key.

There are four options proposed within the menu OPTIONS PIANO: F7 or to display the piano, F5 or to play the last twenty five notes, F9 or to erase the piano, Notes, allowing to hear a note played on the piano keyboard displayed.

Menu FUNCTIONS proposes the following options: F1 or to give an access to the hypertext structure that corresponds to the current musical composition process, F2 or to display the statement of the current exercise, F3 or to erase the last symbol placed on the stave, F4 or to link eighths, F5 or to play the melody written on the highest stave, F6 or to play the melody written on the lowest stave, F8 or to end the composition process, to begin the analysis, and to display the relevant comment, and F10 or to end the work session.

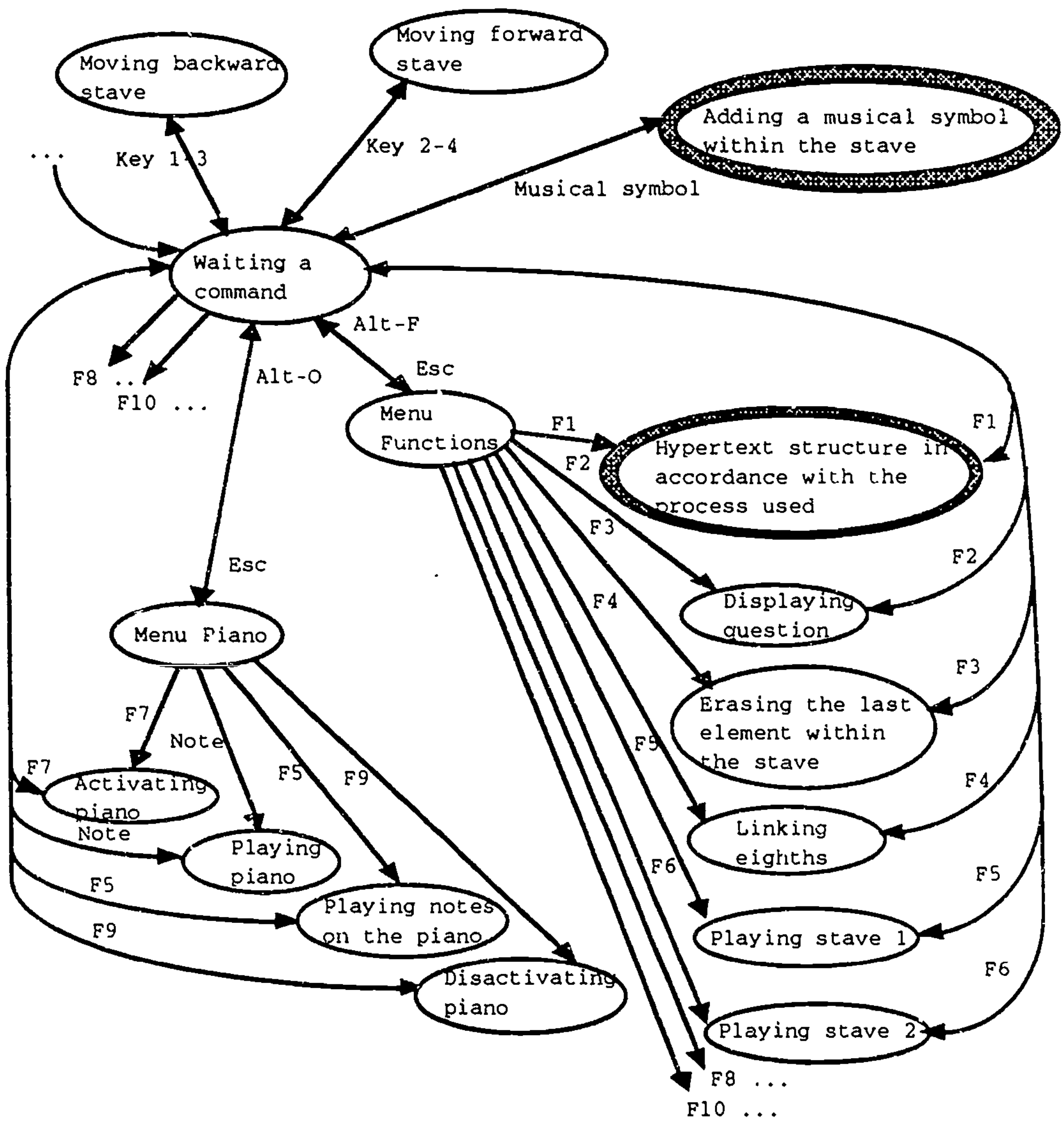


Figure 13.3 Functions allowing to compose.

When a student does an exercise, he or she can press on F1 key to have an access to the prerequisites. An hypertext structure was implemented. It allows a student to browse through prerequisites. Underlined words indicate explanations available. The student can select a word in order to have the explanation related, or can see an index. A word being selected, the explanation related is displayed. The student can either select an underlined keyword, look at the index, come back to the previous window, listen to a melody being part of an explanation, or quit (The SMD related is not illustrated here).

A composition being done, MUSIC analyzes it (figure 13.4). MUSIC checks out if the answer is in accordance with parameters and rules. If so, MUSIC displays a message indicating the success and does comments. If not, weaknesses will be detailed. In both cases, the message has keywords. Depending on his or her needs, the student can brownse through the hypertext structure. One or many examples of composition can also be created by MUSIC; at the student's request, they are commented.

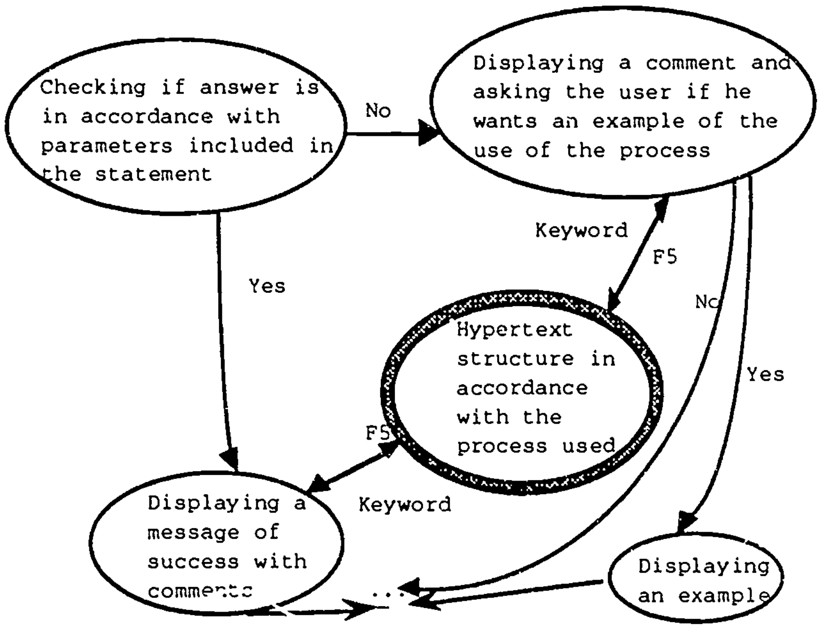


Figure 13.4 Analysis of a composition.

Figure 13.5 shows characteristics of the interface aiming at creating bank and files of exercises. A bank contains all the exercises created by a music teacher in regard of a specific process of composition. A file of exercices contains those which will be done by students in accordance with the spiral curriculum; it is a subset of a bank. To have an access to this part of MUSIC, the music teacher has to press on F9 key when the main menu is displayed. MUSIC waits for a password; at any time, this password can be changed. Three other options are available: to add, to display, and to erase an exercise. When the music teacher indicates that the work session is completed, the main menu is displayed.

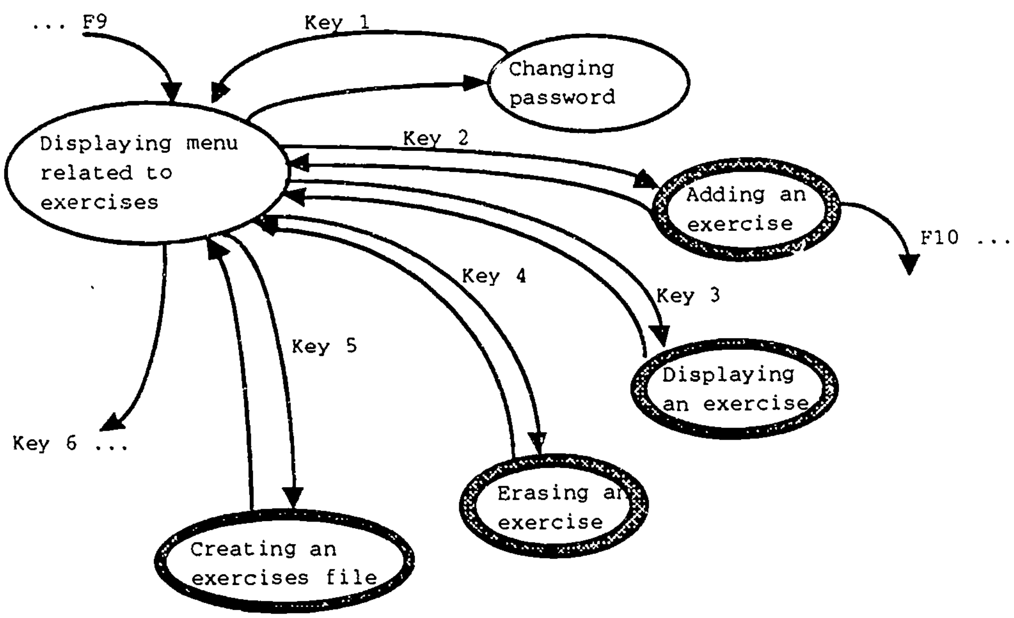


Figure 13.5 Interface allowing to create exercises.

When Adding an exercise is activated (figure 13.6), the music teacher has to identify the process related. He or she also has to provide information such as key signature, time signature, tempo, note and rest figures allowed, etc. The statement of the exercise is then created. All options allowing students to compose are available (figure 13.3). By pressing the key S, the music teacher starts the process of recording the exercise.

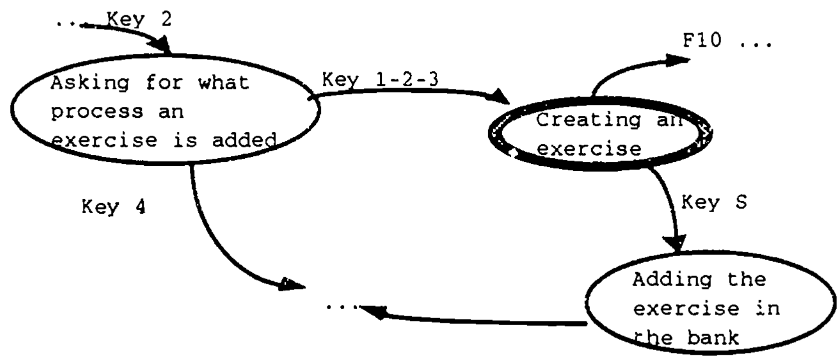


Figure 13.6 Adding an exercise.

In the same way, Displaying an exercise allows the display of exercises of a bank, Erasing an exercise allows to erase the record of an exercise, and Creating a file allows the creation of new files (They are not illustrated here). When Creating a file is selected, the music teacher specifies the musical composition process being concerned. MUSIC can display all linked exercises recorded in the bank.

Communication links between menus

There are seven basic menus (figure 13.7). The following aspects can be found in the main menu (menu 1 at figure 13.7):

1 - Answer (Q-A).
2 - Augmentation, diminution and repetition.
3 - Question - Answer.
4 - Quit.
F9 - Exercises of composition (not displayed).

Depending on the user's selection, either menu 2.1 (Statement of exercices), or menu 2.2 (Composition interface) is activated. The content of menu 2.1 is adjusted in regard of the musical composition process selected. The menu of the interface of composition is made of fixed options when it is used by the music teacher; when it is used by the student, it is made of variable options in regard of the spiral curriculum. The third level has four menus (figure 13.7). Menu 3.0 (Composition analysis) is used at the end of a composition.

Because it presents comments on a composition, its content is variable. The content of menu 3.1 (Help) varies in function of the composition process used. It aims at presenting prerequisites of the composition process being currently used. Links allows to brownse through prerequisites. Menu 3.2 allows to access to one or many of the options of the menu Functions previously described and illustrated at figure 13.3. Menu 3.3 displays options related to the use of the piano keyboard that was described before and illustrated at figure 13.3. Note that only important menus were described here. There are some other secondary menus aiming to dialog with the user.

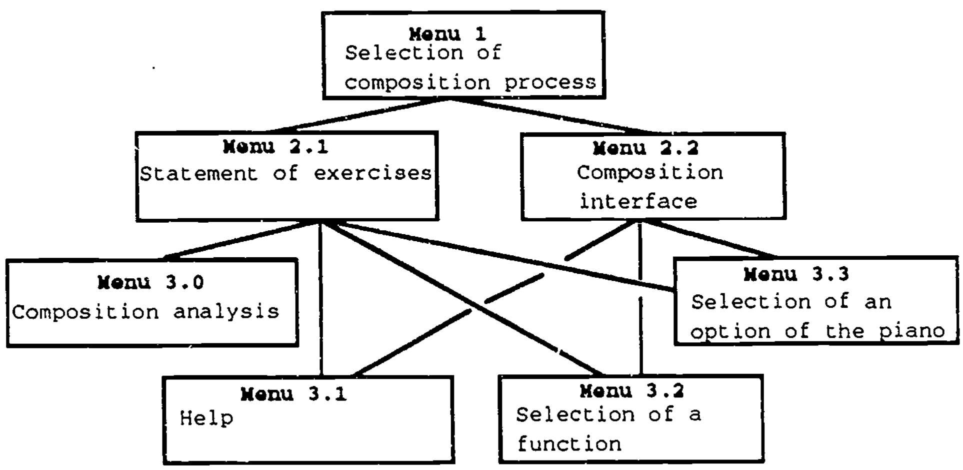


Figure 13.7 Basic menus.

Physical model - Modules hierarchy

The system has been separated into several sub-systems; it is easier to manage it. Modules were conceived considering the main parts of MUSIC, the size and complexity of modules, and a minimum level of interaction between them. The final result is as follows: MUSIC is made of twelve different modules (including four librairies). There are more than ten thousand lines of code.

MUSIC is the module at the top of the hierarchy (figure 13.8). Its first role is to set out the proper graphic mode in regard of the computer being used. It is also used to manage options of the main menu. MUSIC calls MUSICITR module and transmits the user's request. MUSICITR manages the musical composition interface. It calls other modules: MUSICHLP, MUSICANL, and MUSICOPT.

MUSICHLP is called when a student asks for help. The hypertext structure related to the musical composition process being used is then activated. After having considered the extent of the structure, MUSICHLP was separated in two modules: MUSICAS1 and MUSICAS2. MUSICAS1 is used to present explanations while MUSICAS2 is used to present comments issued from the analysis.

MUSICANL module is the one that does the analysis of a composition; MUSICCRI module is also used. MUSICANL establishes determinant elements that must be respected and those that can be used; MUSICCRI checks if they are either respected, or used. The verification being done, MUSICANL calls MUSICHLP module to allow the display of the relevant comment; this call is done mainly because comments refer to keywords. MUSICHLP calls MUSICAS2 that displays the comment, as explained before. The comment being displayed, MUSICANL module takes control and displays, if requested, an example of composition in accordance with parameters mentioned in the statement, and the musical composition process currently used.

MUSICITR calls the MUSICOPT module to have the following functions executed: To display a musical question, To display linked eighths, To erase a symbol on the stave, To play a music score, and To play piano.

The execution of MUSICITR module being ended, MUSIC module takes the control. The main menu is displayed and the process starts again until the user decides to quit.

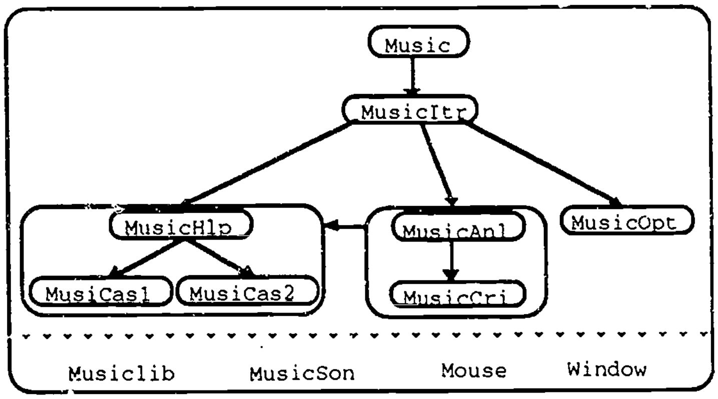


Figure 13.8 Modules hierarchy.

The use of four libraries (those of TURBO PASCAL excluded) was necessary to avoid source code redundancy. These libraries are: MUSICLIB, MUSICSON, MOUSE, and WINDOW.

MUSICLIB contains all the functions allowing the display of music symbols, and the management of files. MUSICSON refers to functions allowing to play and listen a melody. Functions to manage the mouse are included in MOUSE. Functions to manage the windows are included in WINDOW.

In the real files, all the details related to implementation are presented (even the source code). They are not presented here.

Functioning of MUSIC

The main menu allows the selection of the process of composition as illustrated at figure 13.9.

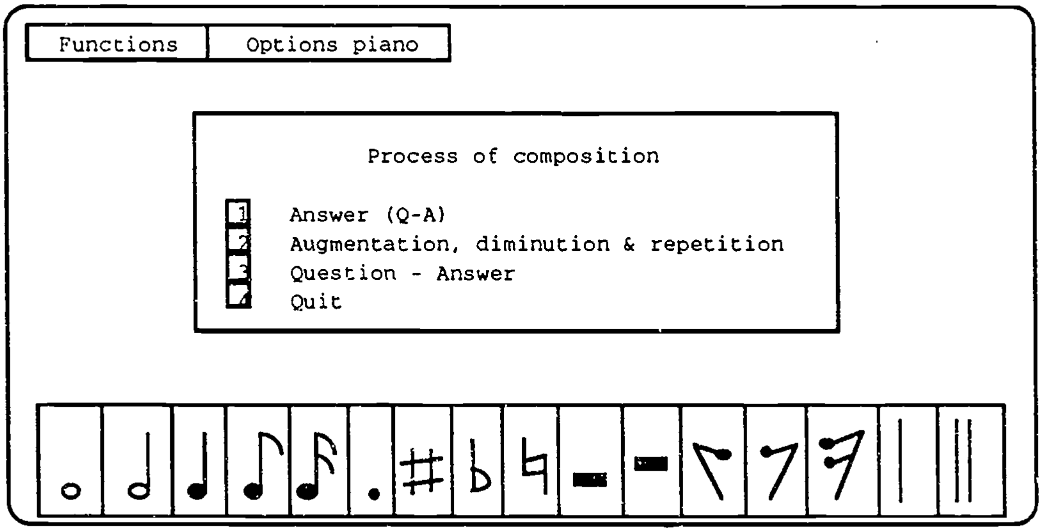


Figure 13.9 Main menu of MUSIC.

The process of composition being selected, the statement of the exercise is displayed. Then, the student is engaged in the process of creating his or her composition. Figure 13.10 illustrates the screen displayed after having asked a student to compose a musical answer to the musical question proposed. There are two staves. The first one proposes the musical question; the second one will be used by the student to compose. When the score is too large to be displayed on one screen, two ends of arrow are displayed. They allow to move the part of score being displayed. They are activated either by selecting one with the mouse, or by typing the associate number on the keyboard. At the bottom of the screen, all musical symbols that can be used by the student in accordance with the spiral curriculum are presented. To select a symbol, the student has to use the mouse or arrows on the numerical keyboard (at the right of the return key).

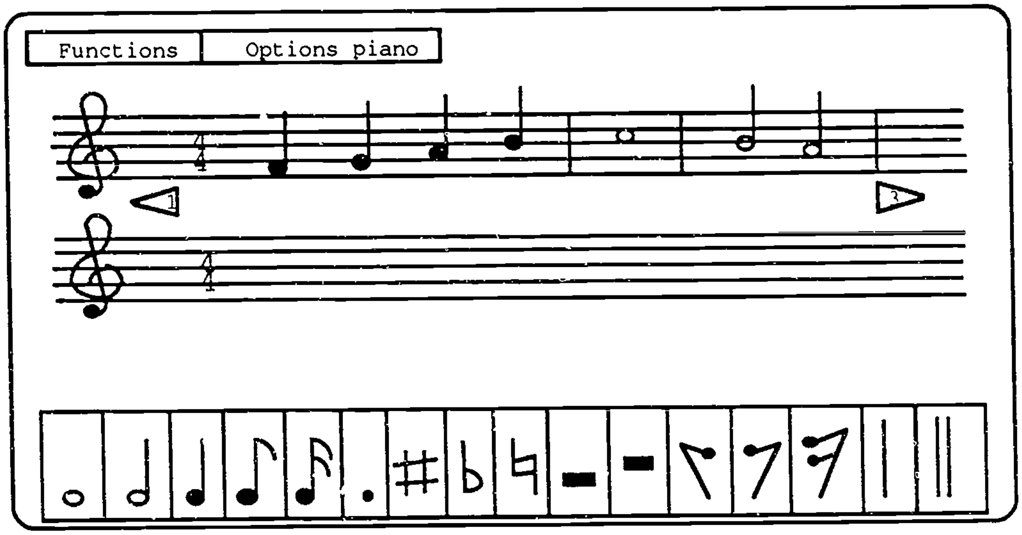


Figure 13.10 Interface.

At the top of the screen presented at figure 13.10, it can be seen that two menus are proposed. One is the Functions menu (figure 13.11).

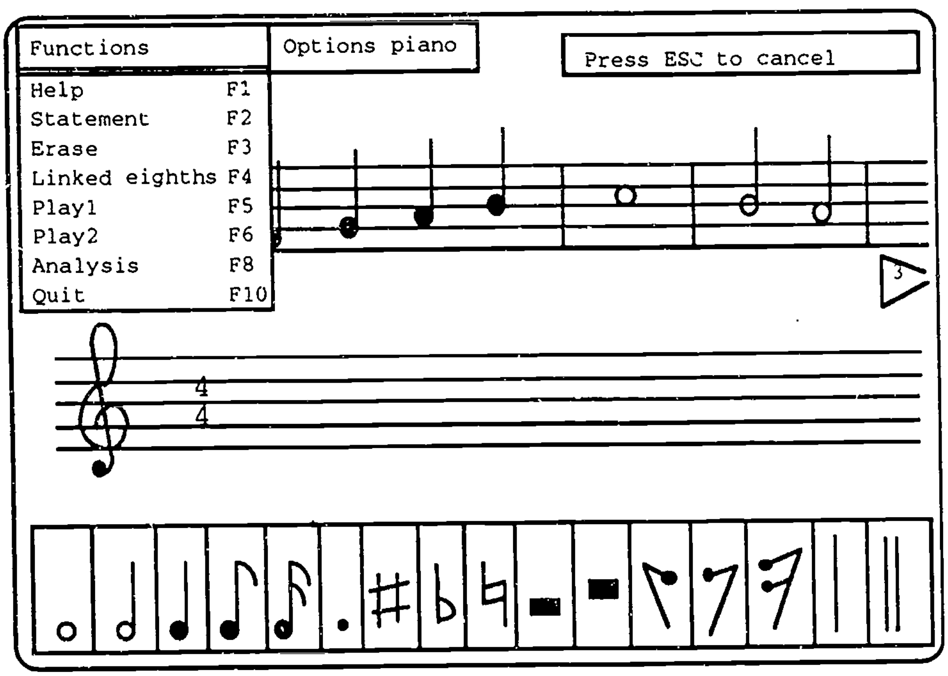


Figure 13.11 Menu presenting functions available.

When a student does exercises, MUSIC can help him or her by presenting explanations related to prerequisites. This help is available at any time; the student merely selects the option Help in the menu Functions, or press on F1 key. An example is presented at figure 13.12; explanation of the concept *accidental* is displayed. Underlined words are keywords. When a keyword is selected, the explanation related is displayed. To select keywords, the mouse or the arrows of the numerical keyboard may be used. The student can see the index, listen to an example related to an explanation (option displayed when there is such an example), come back to the previous window (F5 key), or quit the help option, i.e., to come back to the current composition activity.

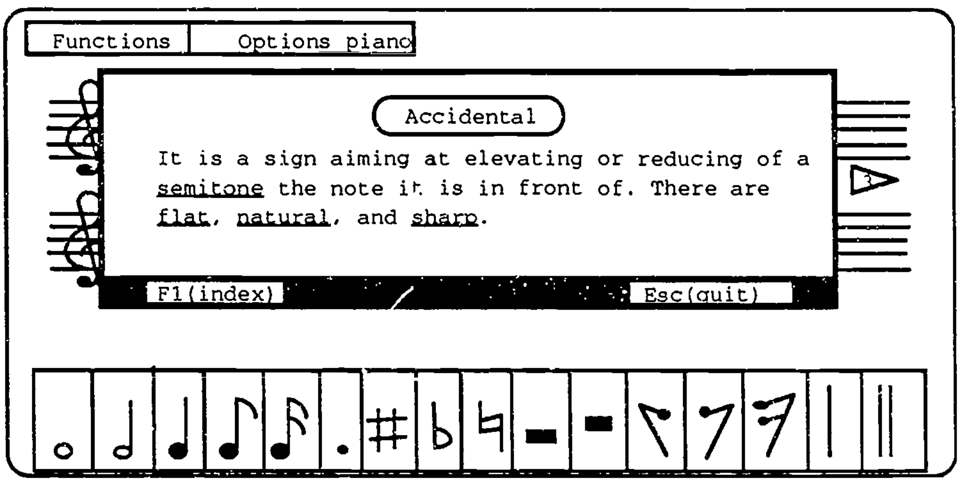


Figure 13.12 Help provided in regard of the accidental concept.

When option Statement is selected (figure 13.11), or when the student presses on F2 key, the statement of the exercise is displayed. To close this window, the student presses on the RETURN key, or clicks in the box included in the window.

Option Erase (figure 13.11) is used to erase the last symbol being placed on the stave. This option is available either by pressing on F3 key, or selecting it in the menu. When there is no symbol on a stave, the access to this option is blocked.

Option Linked eighths (figure 13.11) allows the linking of eighths. When there are not enough eighths that can be be linked, or symbols used are not eighths, the access to the option is blocked. One can also link eighths by pressing on F4 key.

Options Play1 (or F5 key) and Play2 (or F6 key) allow to listen the content of a score in regard of the tempo being selected: Very slow, Slow, Medium, Fast, Very fast. Play1 relates to the highest stave, and Play2 relates to the lowest one.

As illustrated at figure 13.10, the other menu is Options piano (figure 13.13). It can be seen that some options are not always available.

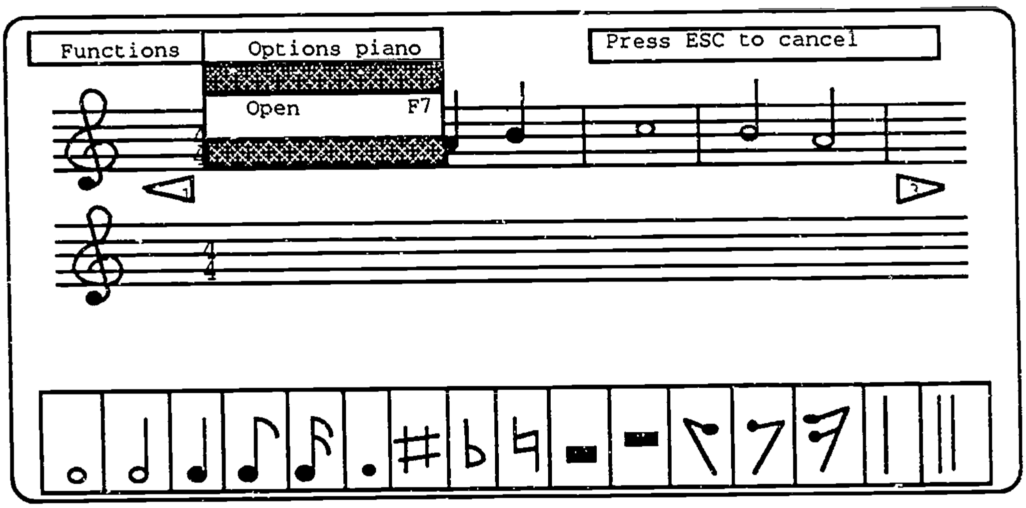


Figure 13.13 Partial content of the menu Options piano.

One of the options not illustrated at figure 13.13 is Use of the piano. It aims at allowing the creation of melody that can be used later to compose. First, the student has to open the piano; either he or she selects the option Open of the menu Options piano (The only one option illustrated at figure 13.13), or the F7 key is pressed on. The piano being opened (and displayed), the student can play on it. The student can type letters and numbers corresponding to each note displayed, or clicks on notes of the piano keyboard using the mouse. A melody being played, the student can hear it again by pressing F5 key or using the menu Options piano. The piano disappears when the F9 is stroken, or the proper option is selected in the menu.

A composition being done, MUSIC analyzes it; that corresponds to the option Analysis of the menu Functions (figure 13.11). Figure 13.14 presents an example. Here again, underlined words refer to prerequisites. MUSIC can also create a composition. If the student clicks on the button Yes, MUSIC will compose. To have a comment on this composition, the student will use the same options that were used to have a comment on his or her own composition.

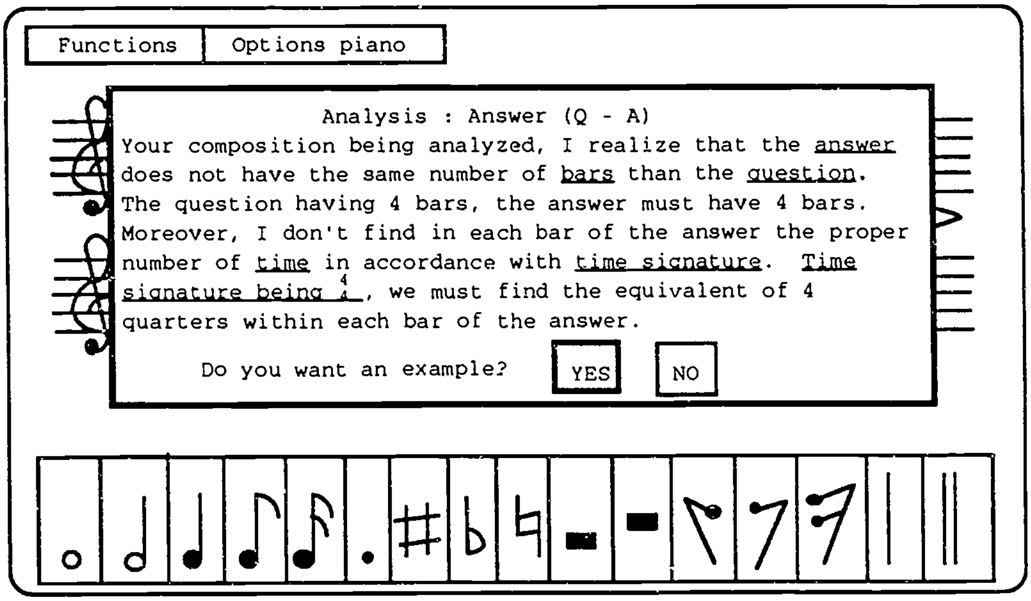


Figure 13.14 Result of an analysis done by MUSIC.

At any time, the student can quit by pressing on F10 key, or by selecting the option Quit in the menu Functions (figure 13.11).

Exercises of composition

In this part, options related to the creation and updating of exercises are summarized. First, the music teacher must press on F9 key when the main menu (figure 13.9) is displayed. MUSIC asks for a password. The music teacher can change his or her password using the option Changing password (figure 13.15). If he or she forgot it, a special program called LECTURE can be used. The current password will then be displayed.

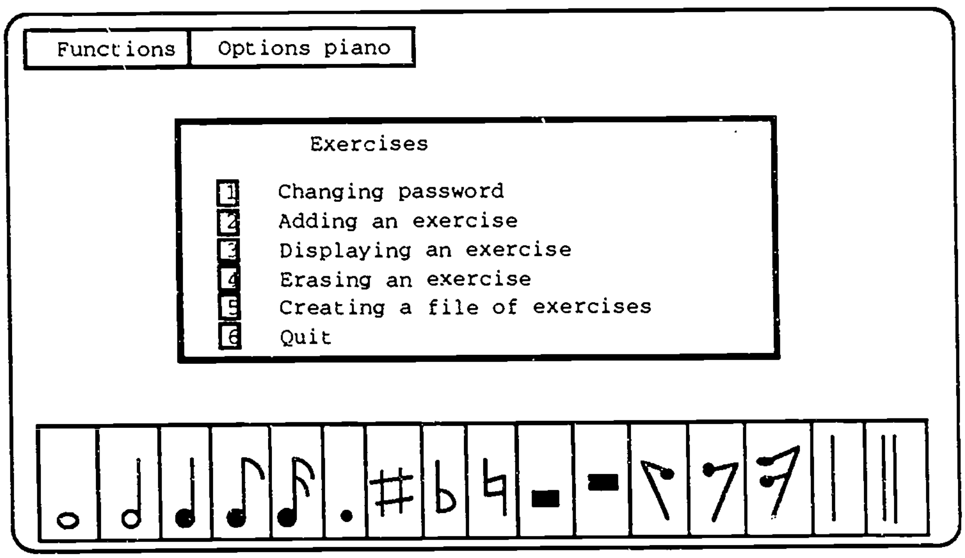


Figure 13.15 Main menu available to the music teacher.

CHAPTER FOURTEEN

Conclusion

To design and develop an intelligent advisor system, it is to plan an environment of help aiming at furthering the use, so the transfer, of several knowledge that an individual previously learned using any instructional method. As an advisor intervenes as an individual using a software application is performing a complex transfer task, its interventions concern the real context of use of the knowledge. An intelligent advisor is an environment of help utterly linked to the use of what has been learned. Users think and integrate several kinds of knowledge to find the "right" way to solve a complex problem. In that sense, an intelligent advisor favors the transfer of knowledge.

To design such an environment, the best set of tactics for presenting to users of an application software, concepts, principles, and rules, being part of a complex transfer task, must be identified. This set of tactics aims at favoring the emergence of the best solution. To favor it, this instructional technology assists both the thinking process of an individual facing a complex problem, the planning process of the appropriate solution, the process of carrying out the plan, and the process of validating the solution. While the advisor is helping, it collects data to be able to adapt the content of its help transactions to the user, and to propose a relevant remedial help.

An intelligent advisor system is a technology being used after initial learning of concepts, principles, and rules. Users of the advisor need to learn them to be able to understand situations in which what is learned is used; they also need to learn those concepts, principles, and rules, to know how to operate in those situations. Any individual needs to know a multitude of facts. Any individual also needs to know how to use them. He or she needs to know when to use them. Having this knowledge, this person needs to learn how to put concepts, facts, performance-skill, principles, rules, and situations together. The orientation assistance to the performance constrasts with the important teaching that focuses on formal and general knowledge. An intelligent advisor does not aim at replacing the formal teaching; it addresses the transfer of knowledge problem. To have transfer of knowledge, something must have been learned. Therefore, an intelligent advisor is used after the formal and general teaching, to favor the transfer of knowledge in real and complex situations. This way of describing the role of an advisor has serious implications for the kind of tactics that are used. Tactics must be highly adaptative and interactive.

The design of a computer intelligent-assistance-to-complex-task system is a substantial orchestration problem. The advisor must be able to consider the process and progress of learning, not merely of simple cases as those proposed to students as exercises, but of very complex tasks. The design and development of an intelligent advisor require analysis of skill and dividing it into manageable and appropriate chunks for the advisor to use. Sophisticated techniques are required for using environmental resources to elaborate this particular help environment. The design of such an environment involve substantial efforts in analysis and specification of knowledge, and decisions taken in regard of the target users. The primary issue is specifying the help environment, i.e. the advisor/user transactions. Decisions must always be taken in regard of the target users of the advisor. After that, come decisions about implementing the design, using computer or multimedia technology. Therefore, technical aspects do not have any influence on the strategy of intervention.

Designing and developing an intelligent advisor involve to join a group of persons coming from various fields. As the development life cycle proposed here divides it into steps, the head of project can plan a way to face this situation.

Table 14.1 presents the whole development life cycle.

Further researches

The author is very concerned about further researches. She presents some suggestions in the following paragraphs.

The approach presented in this book involves the production of a detailed description of the nature of interventions in regard of the target users. Those details are the strategy of intervention. Therefore, when doing rigorously controlled experiments, it is not the effects of the technology "intelligent advisor" that must be measured, but those of a set of strategies and tactics used by an advisor developed for a particular group of users, in regard of a particular transfer task performed in a particular context. Several experiments can be setted, aiming at comparing different combinations of variables in regard of the strategy of intervention. As example, the aim and the target users of an advisor being kept equal, only the way used by the advisor to adapt itself to the user may vary, or only the content of interventions, or only the interface, etc. Suggestions based on research evidence can be made on the variable or group of variables producing the best effects, a same aim and same target group of users being considered. Keeping constant these variables, target users may vary to assess if the effects may be generalized.

Table 14.1

Summary of the development life cycle

	Step 1. Describing the strategy of intervention
1.1	To state the aim of the advisor.
1.2	To identify the target users of the advisor.
1.3	To describe the type of resource.
1.4	To describe the main characteristics of the course of interventions.
1.5	To describe the way used by the advisor to adapt itself to the user.
1.6	To describe the overall characteristics of the subject of interventions.
1.7	To describe the overall characteristics of the vocabulary of presentation.
1.8	To describe the knowledge depth.
1.9	To describe the interface.
	Step 2. Describing the knowledge acquisition strategy
2.1	To select and describe means to delimit the boundaries of the domain.
2.2	To select and describe means and sources of knowledge in regard of the difficulties.
2.2.1	To select and describe means to insure quality of the data collected (difficulties).
2.2.2	To select and describe means to evaluate utility and risk (difficulties).
2.3	To select and describe means and sources of knowledge in regard of the interventions.
2.3.1	To select and describe means to insure quality of the data collected (interventions).
2.3.2	To select and describe means to evaluate utility and risk (interventions).
2.4	To set out the knowledge acquisition method.
	Step 3. Describing the content, the architecture, and the implementation in the environment strategy
3.1	To analyze sources of content.
3.1.x	*Steps of the knowledge acquisition method.*
3.2	To elaborate the content of interventions.
3.2.x	*Steps of the knowledge acquisition method.*
3.3	To produce the architecture of the advisor.
3.3.x	*Steps of the systems analysis method selected.*
3.4	To document the architecture.
3.4.x	*Format of the systems analysis method selected.*
3.5	To describe the environment implementation strategy.
	Step 4. Getting the advisor to the operational stage
4.1	To convert logical models into physical models.
4.1.x	*Steps of the method selected.*
4.2	To validate the content of interventions.
4.3	To program the advisor.
4.4	To test the advisor.
4.4.x	*Steps of the method selected.*
4.5	To implement the advisor.

Utility can also be evaluated; no instructional strategy and tactic can aim at being successful for 100% of its target learners. But, an advisor, aiming at helping users U performing a complex task Y, was useful to many? An organization having a total quality program, keeping records of its workers' errors, could take the decision to invest in the intelligent advisor technology by evaluating if an investment of M$, aiming at helping P% of its workers, and avoiding R% errors, is profitable.

Tactics to implement advisors in their environment is a very important and interesting issue, but particularly neglected. Several questions still stay without answers based on research evidence. Does same advisor being implemented in an equivalent environment have the same efficacy? What are tactics allowing to produce the best effects when an advisor is installed? What are tactics avoiding that the advisor is not used anymore after a certain period of time?

Intelligent advisor systems, a technology with few research evidence suggestions. The design and development of an intelligent advisor, a very complex problem. Several issues of research, and many of them on invisible parts. It is a quite complex future, but the author believes that human can face any complex situations if it is required, and recognized as valuable.

References

Ad Lib (1987). *Ad Lib composition projects #1: User guide.* Québec: Ad Lib Inc.

Airasian, P.W., & Madaus, G.F. (1972). Functional types of student evaluation. *Measurement and Evaluation in Guidance.* 4, 221-233.

Anderson, L., & Block, J. (1977). Mastery learning. In D. Treffinger, J. Davis, & R. Ripple (Eds.), *Handbook of Teaching Educational Psychology.* New York: Academic Press.

Anderson, R.C., & Kulhavy, R.W. (1972). Imagery and prose learning. *Journal of Educational Psychology.* 63, 242-243.

Aubert, A. (1988). *Le solfège et les bases de l'harmonie.* Paris: Éditions Josette Lyon.

Ausubel, D.P. (1960). The use of advance organizer in the learning and retention of meaningful verbal material. *Journal of Educational Psychology.* 51, 267-272.

Ausubel, D.P. (1968). *Educational psychology: A cognitive view.* New York: Holt, Rinehart and Winston.

Barbier, J.M. (1985). *L'évaluation en formation.* Paris: Presses Universitaires de France.

Block, J.H. (1977). Individualized instruction: A mastery learning perspective. *Educational Leadership.* 34, 337-341.

Block, J.H., & Anderson, L.W. (1975). *Mastery learning in classroom instruction.* New York: MacMillan.

Bloom, B.S. (1974). An introduction to mastery «learning theory». In J.H. Block (Ed.), *Schools society, and mastery learning.* New York: Holt, Rinehart and Winston.

Bloom, B.S. (1978a). Change in evaluation methods. In R. Glaser (ed.), *Research and Development and School Change.* New Jersey: Hillsdale.

Bloom, B.S. (1978b). New views of the learners: Implications for instruction and curriculum. *Educational Leadership.* 35, 563-576.

Bloom, B.S. (1979). *Caractéristiques individuelles et apprentissages scolaires.* (Traduit de l'américain par Viviane de Landsheere). Bruxelles: Labor.

Bloom, B.S., Hastings, J.T., & Madaus, G.F. (1971). *Handbook on formative and summative evaluation of student learning.* New York: McGraw Hill.

Bloom, B.S., Madaus, G.F., & Hastings, J.T. (1981). *Evaluation to improve learning.* New York: McGraw Hill.

Bonboir, A. (1970). *La pédagogie corrective.* Paris: Presses Universitaires de France.

Boulet, M.M. (1988a). *Système conseiller: Analyse de l'objectif d'unité 1 "Démarrer un processus de modélisation conceptuelle des données".* Rapport de recherche DIUL-RR-8815. Québec: Département d'informatique, Faculté des Sciences et de génie, Université Laval.

Boulet, M.M. (1988b). *Système conseiller: Analyse de l'objectif d'unité 2 "Identifier et représenter à l'aide du formalisme Merise les individus d'un modèle conceptuel de données, leur identifiant et leur liste de propriétés".* Rapport de recherche DIUL-RR-8816. Québec: Département d'informatique, Faculté des Sciences et de génie, Université Laval.

Boulet, M.M. (1988c). *Système conseiller: Analyse de l'objectif d'unité 3 "Identifier et représenter à l'aide du formalisme Merise les relations d'un modèle conceptuel de données, leur identifiant et leur liste de propriétés".* Rapport de recherche DIUL-RR-8817. Québec: Département d'informatique, Faculté des Sciences et de génie, Université Laval.

Boulet, M.M., & Barbeau, L. (1988). *Système conseiller: Analyse de l'objectif d'unité 4 "Identifier et représenter à l'aide du formalisme Merise les cardinalités et contraintes d'un modèle conceptuel de données".* Rapport de recherche.

Québec: Département d'informatique, Faculté des Sciences et de génie, Université Laval.

Boulet, M.M., & Dufour, F. (1991). *Système conseiller en composition musicale. Le procédé question-réponse: Analyse des connaissances requises en troisième secondaire pour composer un thème mélodique en utilisant le procédé question-réponse*. Rapport de recherche DIUL-RR-9107, Québec: Département d'informatique, Faculté des Sciences et de Génie, Université Laval.

Boulet, M.M., Dufour, F., & Lavoie, L. (1991a). *Système conseiller en composition musicale. Le procédé question-réponse: Analyse des connaissances requises en troisième secondaire pour composer une réponse à une question en considérant le programme de musique du secondaire*. Rapport de recherche DIUL-RR-9102. Québec: Département d'informatique, Faculté des sciences et de génie, Université Laval.

Boulet, M.M., Dufour, F., & Lavoie, L. (1991b). *Système conseiller en composition musicale. Les procédés d'augmentation, de diminution et de répétition: Analyse des connaissances requises en troisième secondaire pour développer un thème mélodique à l'aide de ces procédés en considérant le programme de musique du secondaire*. Rapport de recherche DIUL-RR-9105, Québec: Département d'informatique, Faculté des Sciences et de Génie, Université Laval.

Boulet, M.M., Labbé, P., Laprise, Y., Lemay, F., & Lavoie, L. (1987). Conception de systèmes de formation informatisés à base de connaissances selon une approche diagnostique. *IEEE Montech '87 - Compint '87*. Montréal, 70-75.

Boulet, M.M., Labbé, P., Lavoie, L., Lemay, F. (1989). Educational knowledge based system design using a diagnostic approach. In H. Maurer (Ed.), *Proceedings of International Conference on Computer Assisted Learning*. Dallas, 1-13.

Boulet, M.M., Lavoie, L., Slobodrian, S., & Tang, J. (1989). *Système conseiller: description du module d'explication du domaine conçu à partir de la théorie d'Ausubel et les schémas organisateurs.* Rapport de recherche DIUL-RR-8906. Québec: Département d'informatique, Faculté des Sciences et de génie, Université Laval.

Brien, R. (1980). *Design pédagogique: Introduction à l'approche de Gagné et Briggs.* Sainte-Foy: Les Éditions Saint-Yves.

Brown, J.S., Burton, R.R., & De Kleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J.S. Brown (eds), *Intelligent Tutoring Systems.* London: Academic Press.

Bruner, J. (1962). *On knowing.* Cambridge: Harvard Univ. Press.

Bruner, J. (1963). *The process of education.* New York: Vintage Books.

Bruner, J. (1966a). *Studies in cognitive growth.* New York: Wiley.

Bruner, J. (1966b). *Toward a theory of instruction.* Cambridge: Harvard Univ. Press.

Buchanan, B.G., & Shortliffe, E.H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project.* Reading: Addison-Wesley.

Burns, M.T. (1988). Music as a tool for enhancing creativity. *The Journal of Creative Behavior.* 22, 62-69.

Burton, R.R., & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J.S. Brown (eds), *Intelligent Tutoring Systems.* London: Academic Press.

Caponigri, R.S. (1980). The development of a large-scale mastery learning program in a community college system. *First National Mastery Learning Conference.* Chicago, 35-37.

Cheek, M.C., & Cheek, E.H. (1980). *Diagnostic-prescriptive reading instruction: A guide for classroom instruction*. Iowa: WCP.

Chen, P.P. (1976). The entity relationship model: Toward a unified view of data. *ACM Transactions on database systems*. 1, 9-36.

Clancey, W.J. (1982). Tutoring rules for guiding a case method dialog. In D. Sleeman & J.S. Brown (eds), *Intelligent Tutoring Systems*. London: Academic Press.

Coleman, E.B. (1965). Responses to a scale of grammaticalness. *Journal of Verbal Learning and Verbal Behavior*. 4, 521-527.

Crighton, A. (1980). *A workbook for music analysis*. Waterloo: Waterloo Music.

Downey, R.G., & Hakes, D.T. (1968). Some psychological effects of violating linguistics rules. *Journal of Verbal Learning and Verbal Behavior*. 7, 158-161.

Dwyer, F.M. (1970). Exploratory studies in effectiveness of visual illustrations. *AV Communication Review*. 18, 235-249.

Egan, D.E., & Greeno, J.G. (1973). Acquiring cognitive structure by discovery and rule learning. *Journal of Educational Psychology*. 64, 85-97.

Fontaine, F. (1979). *Dossier sur l'évaluation*. Montréal: Université de Montréal, Services pédagogiques.

Fournier, G., Milot, J., Richard, G., Béchard, A., & De Melo, D. (1986). *Musicontact 3*. Montréal: Les Éditions HRW.

Gagné, R.M. (1962). The acquisition of knowledge. *Psychological Review*. 69, 355-365.

Gagné, R.M. (1968). Learning hierarchies. *Educational Psychologist*.

Gagné, R.M. (1972). Domains of learning. *Interchange*. 3, 1-8.

Gagné, R.M. (1974). *Essentials of learning and instruction*. Hinsdale: The Dryden Press.

Gagné, R.M. (1977). *The Conditions of Learning*. New York: Holt, Rinehart and Winston.

Gagné, R.M., & Briggs, L.J. (1979). *Principles of instructional design*. New York: Holt, Rinehart and Winston.

Gagné, R.M., & Brown, L.T. (1961). Some Factors in the Programming of Conceptual Learning. *Journal of Experimental Psychology*. 62, 313-322.

Gagné, R.M., & White, R.T. (1978). Memory Structures and Learning Outcomes. *Review of Educational Research*. 48, 187-222.

Gagné, R.M., Mayor, J.R., Garstens, H.L., & Paradise, N.E. (1962). Factors in Acquiring Knowledge of a Mathematical Task. *Psychological Monographs*. 76, 1-19.

Gane, C., & Sarson, T. (1979). *Structured systems analysis: Tools and techniques*. Englewood Cliffs: Prentice-Hall Inc.

Glaser, R. (1963). Instructional technology and the measurement of learning outcomes: Some questions. *American Psychologist*. 18, 519-521.

Goetschius, P. (1920). *Exercices in melody writing*. New York: Schirmer Inc.

Guéladé, F. (1987). *Effets sur l'apprentissage et la rétention d'habiletés intellectuelles du rappel des schémas organisateurs comme préalables*. Thèse inédite de doctorat, Québec: Université Laval.

Guttman, L., & Schlesinger, I.M. (1967a). Systematic construction of distructors for ability and achievement test items. *Educational and Psychology Measurement*. 27, 569-580.

Guttman, L., & Schlesinger, I.M. (1967b). *The analysis of diagnostic effectiveness of a facet design battery of achievement and analytical ability tests*. Jerusalem: The Israel Institute of Applied Social Research.

Hartley, J. (1985). *Designing instructional text*. London: Kogan Page.

Hill, W.C., & Miller, J.R. (1988). Justified advice: A semi-naturalistic study of advisory strategies. *Proceedings of CHI'88: Human Factors in Computing Systems*. Washington D.C., 185-190.

Hively, W. (1974). Introduction to domain-referencing testing. *Educational Technology*. 14, 5-10.

Jones, G.T. (1963). *Music composition: A manual for training the young composer*. Evanston: Summy-Birchard Co.

Kintsch, W. (1972). Notes on the structure of semantic memory. In E. Tulving and W. Donaldson (Eds.), *Organization of Memory*. New York: Academic Press.

Kraft, L. (1976). *Gradus I: An integrated approach to harmony, counterpoint, and analysis*. New York: W.W. Norton & Company Inc.

Landa, L.N. (1976). *Instructional regulation and control*. Englewood Cliffs: Educational Technology Publications.

Leshin, C.B., Pollock, J., & Reigeluth, C.M. (1992). *Instructional design strategies and tactics*. Englewood Cliffs: Educational Technology Pub.

Manhattanville Music Curriculum Program. (s.d.). *MMCP synthesis*. Bardonia: Media Materials Inc.

Marks, L.E., & Miller, G.A. (1964). The role of semantic and syntactic constraints on the memorization of English sentences. *Journal of Verbal Learning and Verbal Behavior*. 3, 1-5.

Mayer, R.E. (1974). Acquisition and resilience under varying testing conditions for structurally different problem-solving procedures. *Journal of Educational Psychology*. 66, 644-656.

Mayer, R.E., Stiehl, C.C., & Greeno, J.G. (1975). Acquisition of understanding and skill in relation to subjects' preparation and meaningful of instruction. *Journal of Educational Psychology*. 67, 331-350.

McCoy, K.F. (1983). Correcting misconceptions: what to say when the user is mistaken. *Proceeedings of CHI'83: Human Factors in Computing Systems*. Washington D.C., 197-201.

Merrill, M.D. (1988). Applying component display theory to the design of courseware. In D.H. Jonassen (Ed.). *Instructional designs for microcomputer courseware*. Hillsdale: Lawrence Erlbaum Ass.

Ministère de l'Education. (1981a). *Politique générale d'évaluation pédagogique. Secteur du préscolaire, du primaire et du secondaire*. Québec: Gouvernement du Québec, Direction générale du développement pédagogique, Direction de l'évaluation pédagogique.

Ministère de l'Education. (1981b). *Programme d'études secondaires. Musique*. Québec: Bibliothèque Nationale du Québec.

Ministère de l'Education. (1983). *Guide pédagogique: Secondaire: Musique*. Québec: Gouvernement du Québec, Ministère de l'éducation, Direction générale du développement pédagogique.

Moore, B.R. (1990). The relationship between curriculum and learner: Music composition and learning style. *Journal of Research in Music*. 38, 24-38.

Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs: Prentice-Hall.

Nisbett, R., & Wilson, T. (1977). Telling more than we know: Verbal reports on mental processes. *Psychological review*. 84, 231-259.

Noizet, G., & Caverni, J.P. (1978). *Psychologie de l'évaluation scolaire*. Paris: Presses Universitaires de France.

Paivio, A. (1971). *Imagery and verbal process*. New York: Holt, Rinehart and Winston.

Paivio, A. (1975). Coding distinctions and repetition effects in memory. In G.H. Bower (Ed.), *The Psychology of Learning and Motivation*. New York: Academic Press.

Parent, A. (1990). *Collection and analysis of dialogue protocols for the question-answering facility of the entity-relationship modeling advisor*. Ottawa: National Research Council Canada.

Peeck, J. (1974). Retention of pictorial and verbal content of a text with illustration. *Journal of Educational Psychology*. 66, 880-888.

Provencher, G. (1985). Les fonctions informatives et les principaux facteurs des feed-back correctifs dans l'apprentissage scolaire. *Revue des Sciences de l'Education*. 11, 67-81.

Radocy, R. E., & Boyle, J. D. (1987). *Measurement and evaluation of musical experiences*. Macmillan Inc. : Schirmer Books.

Rasco, R.W., Tennyson, R.D., & Boutwell, R.C. (1975). Imagery instructions and drawings in learning prose. *Journal of Educational Psychology*. 67, 188-192.

Reigeluth, C.M. (1987). Lesson blueprints based on the elaboration theory of instruction. In C.M. Reigeluth (Ed.), *Instructional theories in action: Lessons illustrating selected theories and models*. Hillsdale: Lawrence Erlbaum Ass.

Reigeluth, C.M., & Stein, F.S. (1983). The elaboration theory of instruction. In C.M. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status*. Hillsdale: Lawrence Erlbaum Ass.

Rich, E. (1983). Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies*. 18, 199-214.

Richaudeau, F. (1975). *Les secrets de la communication efficace*. Paris: Retz-CEPL.

Richaudeau, F. (1979). *Le langage efficace*. Paris: Marabout.

Robidas, G. (1990). *Psychologie de l'apprentissage: Un système d'apprentissage-enseignement personnalisé*. Brossard: Behaviora Inc.

Rubinstein, R., & Hersh, H. (1984). *The human factor: Designing computer systems for people*. Mass.: Digital Press.

Scallon, G. (1988a). *L'évaluation formative des apprentissages: L'instrumentation*. Québec: Presses de l'Université Laval.

Scallon, G. (1988b). *L'évaluation formative des apprentissages: La réflexion*. Québec: Presses de l'Université Laval.

Schoenberg, A. (1987). *Fondements de la composition musicale*. Paris: JCLattès.

Tardieu, H., Nanci, D., & Pascot, D. C. (1984). *Conception d'un système d'information: construction de la base de données*. Chicoutimi: Gaëtan Morin éd.

Tardieu, H., Rochfield, A., & Coletti, R. (1983). *La méthode Merise: Principes et outils*. France: Les Éditions d'Organisation.

Warburton, A.O. (1960). *Melody writing and analysis*. London: Longmans.

Weigand, V.K. (1969). *A study of subordinate skills in science problem solving*. Ph.D. Dissertation, Berkeley: University of California.

White, R.T. (1869). *The composition of simple melodies*. London: Novello and Co. Ltd.

ISBN 2-9802934-1-5