Cogn Tech Work (2003) 5: 248–271
DOI 10.1007/s10111-003-0128-9

ORIGINAL ARTICLE

S. Lepreux · M. Abed · C. Kolski

# A human-centred methodology applied to decision support system design and evaluation in a railway network context

Received: 14 June 2002 / Accepted: 12 May 2003 / Published online: 26 July 2003
© Springer-Verlag London Limited 2003

**Abstract** This article describes a methodological process for the design and evaluation of an interactive system in an industrial context. The process is called the U-model, and has been used in many projects since the early 1990s. The article describes a recent case study in which the U-model was adapted and used during a project involving the design of a decision support system intended for a railway investment context.

**Keywords** Decision support system · Design · Evaluation · Human-machine interface · Interactive system · Methodology

## 1 Introduction

For many years now, researchers in human-machine interaction have been working on methodological processes which can be used for the design and evaluation of the interactive systems found in a human-machine system context (Singleton 1974; Woods 1986; Abed et al. 1991; Millot and Debernard 1993; Kolski and Millot 1991; Kolski 1997; Helander et al. 1997; Hollnagel and Cacciabue 1999; Millot 1988; Moussa et al. 2000).

Software design models and methods are also available in software engineering. These models often prove to be rather unsuitable when the system in question is interactive. For example, the notions of analysis and modelling of human tasks and characteristics, along with the notion of ergonomic evaluation of the software, are not dealt with.

Consequently, the first part of this article deals with the main propositions made in the software engineering field. An overall critical view is developed, and then several human-computer interaction (HCI) enriched models are reviewed. The second part of the article is an in-depth description of a process, called the U-model, which we originally proposed at the beginning of the 1990s and which has now been improved and progressively validated during many industrial projects. This process takes its source from the models described previously, whether or not they have been enriched from the human-machine interaction angle.

In the third part, we describe a case study called INFRAFER, which was part of a project sponsored by the French Ministry of Education, Research and Technology, and which involved three partners: RFF (Réseau Ferré de France, Paris), CORYS TESS (Grenoble), and the LAMIH. During this project, which was aimed at the design and evaluation of an interactive decision support system to be used in a railway investment context, we based our studies on a methodological process adapted from the U-model.

## 2 The limitations of the software engineering models and HCI enriched cycles

The aim of a software development model (or cycle) is to specify the logical or temporal order in which the stages to produce a software programme happen, whether the software is interactive or not. Over the past fifteen years, there has been an important move away from the classic cycles of software engineering such as the Waterfall models, the V-models and the spiral and incremental models (and their variations), towards cycles which integrate the human dimension and greatly favour prototyping.

Before examining the properties of these human-machine oriented design cycles which we will call *HCI enriched models*, we will give a brief description of the classic cycles according to the manner in which they

S. Lepreux (✉) · M. Abed · C. Kolski
Laboratoire d'Automatique,
de Mécanique et d'Informatique,
Industrielles et Humaines,
Le Mont-Houy University of Valenciennes,
LAMIH-UMR CNRS 8530,
59313 Valenciennes, France
E-mail: sophie.lepreux@univ-valenciennes.fr
Fax: +33-03-27511316

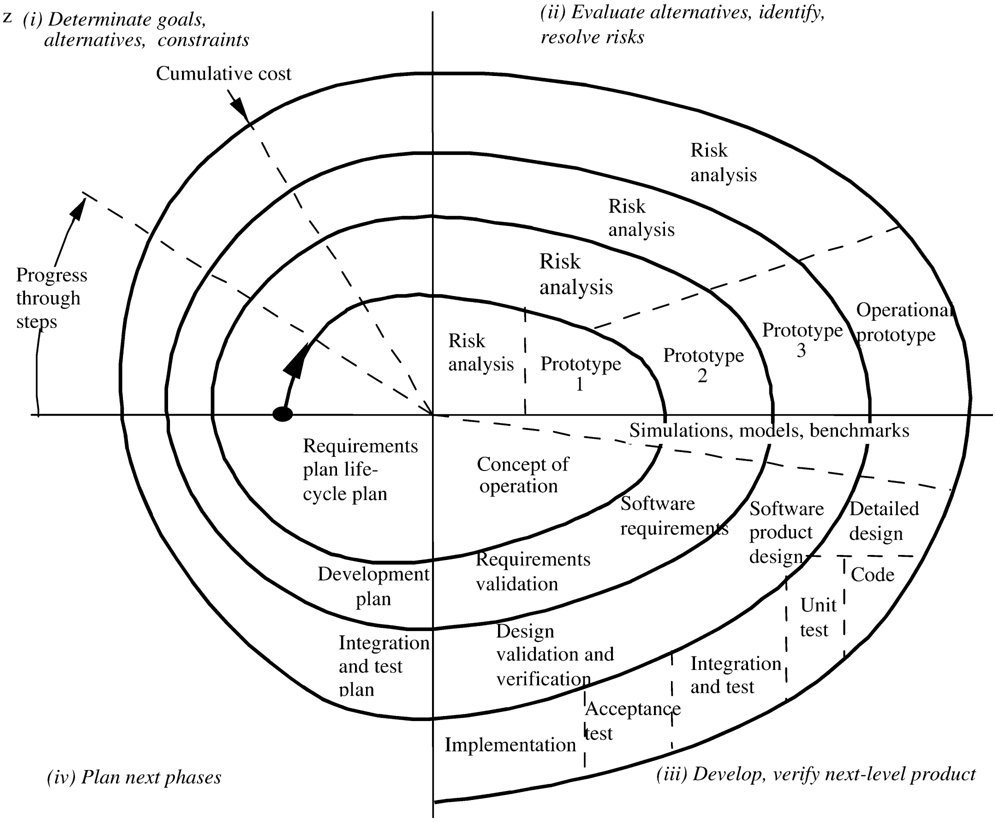


**Fig. 1** A spiral model

implicitly or explicitly deal with human-machine interaction. It should be noted that these cycles are described in detail in many books and articles (Kolski 1997; Sommerville 1994; Thayer and McGettrick 1993).

## 2.1 Classic development cycles provided by software engineering

The waterfall model designed by Boehm (1981), is one of the first models which appeared to meet industrial needs in terms of productivity and software quality. It defines a sequential performance of the development process stages; returns are only possible to the previous stage in order to take any deficiencies identified into account. As far as the development of an interactive system is concerned, no analysis or modelling of the potential user tasks is recommended. In fact, these extremely important notions are considered simply according to the common sense of the most experienced designers and in a mostly informal manner during the first stage. The user aspect is only involved, implicitly, in the final stages of the evaluation of the product developed. It is clear that the waterfall model cannot be adapted, as it is, to suit a problem of interactive application design in which certain general principles such as the analysis of user needs, the user characteristics, the development of prototypes from the first phases of the design process, the iterative evaluation, etc., are very important.

The V model (McDermid and Ripkin 1984) is used in many companies and is recommended by industrial quality promotion organisations. There are different variants of this model (Jaulent 1990; Thayer and McGettrick 1993; Arlat 1995). The V model structures the stages of the cycle, which remain identical globally to those of the waterfall model, into two processes: (i) downward for the specification and design (ii) upward for the validations and tests. The plan, means and methods to evaluate and validate the results of the phase must be included in each phase of the downward approach. This concern with providing for the evaluation of the system as far upstream as possible, and precisely with regard to each phase, is an undeniably strong point of the V model. However, it only provides for very limited returns, which can be a handicap for an iterative design. It should be noted that this model is criticised when the focal point in the development is a software programme with a high interactive content. Indeed, the analysis and modelling of human and user tasks are not situated. However, because of its simplicity and ability to be applied to any application, several authors, including Kolski (1997) and Coutaz (1995), chose it and adapted it as a development framework for interactive applications.

Unlike the first two models, the spiral model introduced by Boehm et al. (1984) represents an *iterative* process (Fig. 1). This model is very interesting for the development of highly interactive software, given that needs are formulated progressively, and the various risks are analysed and resolved as they are encountered. Unlike the previously mentioned models, this one has the advantage of evaluating risks and not

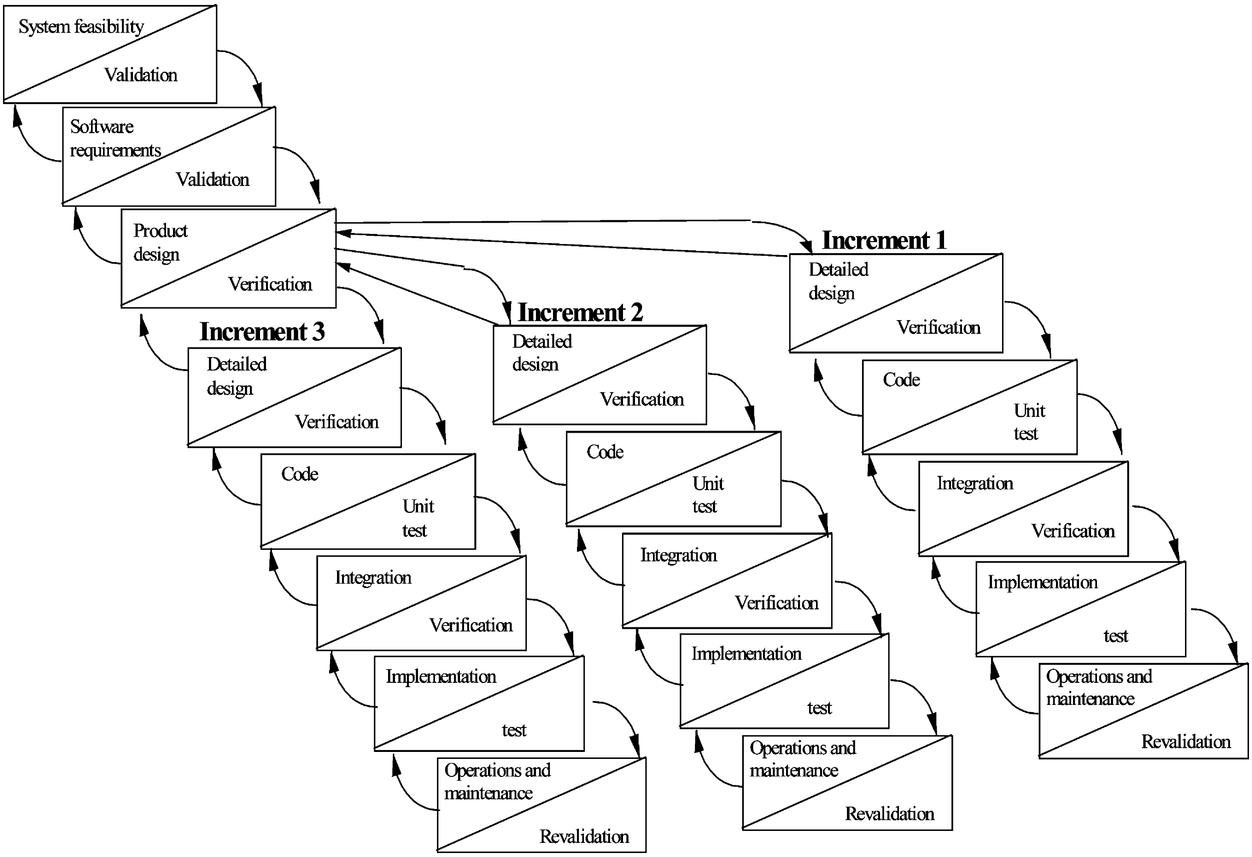


**Fig. 2** An incremental model

beginning the detailed development of other less risky software elements until the high risk elements have been resolved. The other advantage, which would seem to indicate promising perspectives for the development of interactive systems, is that of prototyping, which introduces the evaluation of the solutions envisaged from the beginning of the cycle. The disadvantage of the spiral model is that it does not explicitly integrate the analysis and modelling of the users, even though its process implies them; they are left to the appreciation of the designer, as with the previous models. The spiral model is the same as the incremental model as far as prototyping is concerned: as from a given phase (generally the architectural design phase), the process is iterated several times, resulting each time in the production of increments (Fig. 2). Each increment corresponds to an operational software programme which gets closer each time to the finished product through the addition of functionalities; the evolutions between increments are guided by operational experiments (ESA 1991; Arlat 1995). However, like the other models, the specificities linked to human-machine interaction are not explicitly dealt with and therefore remain at the appreciation of the designer. This model can therefore also be improved.

The traditional models suggested in software engineering are therefore very generic and better suited to the development of software which is not very interactive, or not interactive at all, than to the interactive applications we are concerned with, which must be as useful as they are usable. However, these models remain at the basis of the methods and models used for human-machine interaction, called *HCI enriched models*, which will be dealt with in the following paragraphs.

## 2.2 Enriched cycles for the development of interactive systems

The idea of enrichment modifies several essential aspects of the classic cycles, making it necessary to reconsider their structure and organisation (Kolski et al. 2001). Nevertheless, the models suggested do not necessarily claim to provide the total coverage of a project aimed at the design and development of an interactive system. The main concern of these models is above all to emphasise, from the methodological point of view, fundamental aspects such as the modelling of human tasks, the iterative development of prototypes and the evaluation of the human-machine system. Even though these models have possible limitations, it is interesting to examine some of them and to identify the strong points of each one as regards the problem of interactive systems. These models are to be considered as theoretical and methodological frameworks for an interactive system development process, rather than finished tools.

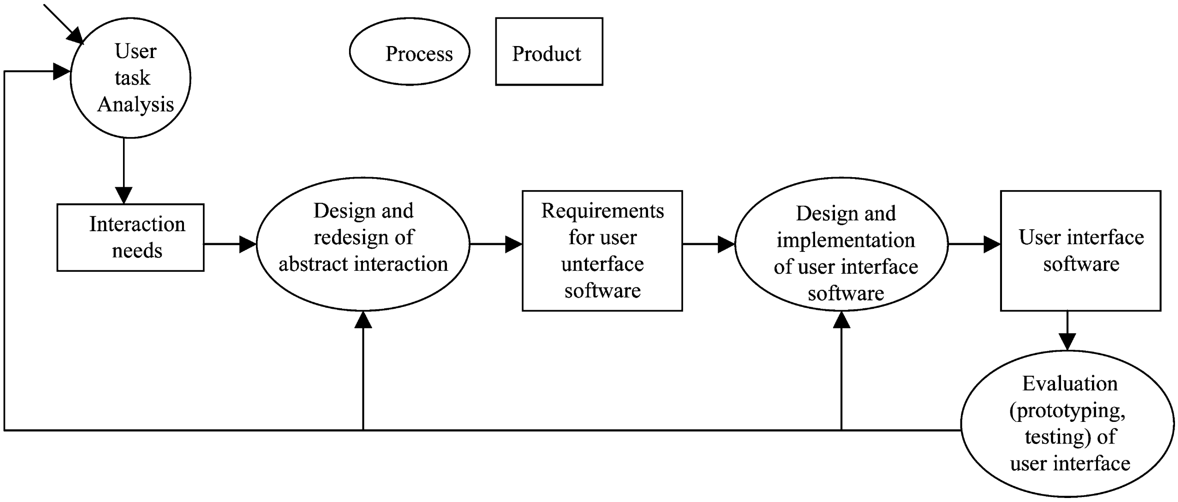

Process
Product
User task Analysis
Interaction needs
Design and redesign of abstract interaction
Requirements for user unterface software
Design and implementation of user interface software
User interface software
Evaluation (prototyping, testing) of user interface

**Fig. 3** A user interface design cycle according to Hartson and Boehm-Davis (1993)

In this section, we concentrate essentially on the enriched models made up of interconnected phases based on classic models from the field of software engineering. This is why we have chosen in this paper not to consider methods and models based on a theory (or elements of a theory) of human interaction with machines and the environment.

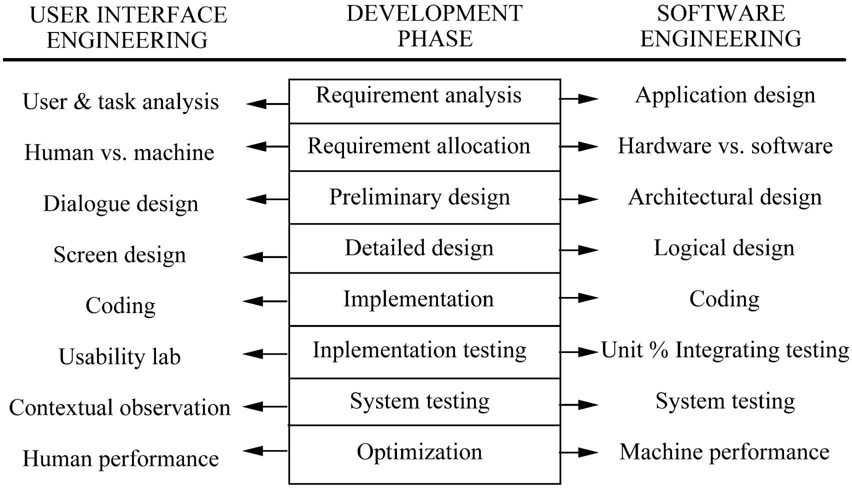
As an example, the model developed by Hartson and Boehm-Davis (1993) (Fig. 3) makes it possible to integrate particular stages for the development of an interface in an existing software engineering method, selected by the designer. Indeed, this model translates the wishes of authors to divide any development process into a period of specification and a period of implementation, in alternation.

The Curtis and Hefley model (1994) merits careful consideration for each of the classic stages of software engineering. The model situates the work to be performed; in the left-hand part of the model around human-machine interaction, and in the right-hand part of the model, around the aspects usually linked to software development (Fig. 4). It therefore specifies the additional tasks which must be performed throughout the project, which can be an extremely useful aspect for project leaders.

The model designed by Hix and Hartson (1993), also called the star model (Fig. 5) situates the evaluation at the very centre of the complete cycle, thus showing possible interactions/iterations between each of the stages. The evaluation stage is seen as an intermediate stage, which makes it possible to protect the development team from an ultimate rejection, which can be seen as a sanction. Even though this model is fairly far from being a classic model, this idea makes it interesting. It does not impose an order in which the stages of the process must be performed, although in practice the development activities are placed at the end of the cycle. It should be noted that it implies a participative design aimed at the early detection of usability problems, requiring a high degree of user implication because of this central idea (Hix and Hartson 1993; Poltrock and Grundin 1995).

The ∇ model (pronounced "nabla") (Kolski 1997, 1998), built following a double V-shaped cycle, situates

**Fig. 4** A model showing user interface engineering/software engineering integration (Curtis and Hefley 1994)



| USER INTERFACE ENGINEERING | DEVELOPMENT PHASE | SOFTWARE ENGINEERING |
|---|---|---|
| User & task analysis ← | Requirement analysis | → Application design |
| Human vs. machine ← | Requirement allocation | → Hardware vs. software |
| Dialogue design ← | Preliminary design | → Architectural design |
| Screen design ← | Detailed design | → Logical design |
| Coding ← | Implementation | → Coding |
| Usability lab ← | Inplementation testing | → Unit % Integrating testing |
| Contextual observation ← | System testing | → System testing |
| Human performance ← | Optimization | → Machine performance |

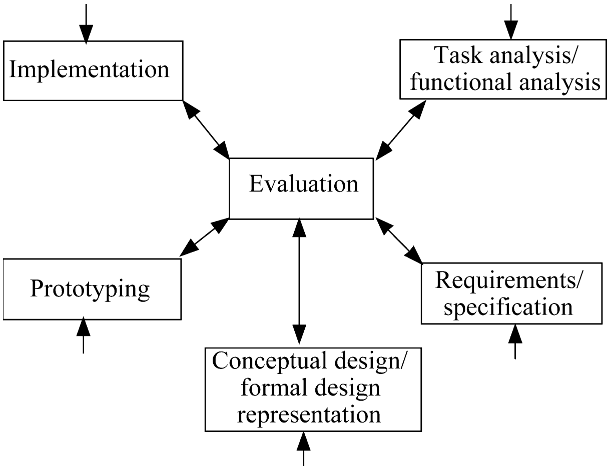

Implementation
Task analysis/ functional analysis
Evaluation
Prototyping
Requirements/ specification
Conceptual design/ formal design representation

**Fig. 5** A star model (Hix 1995)

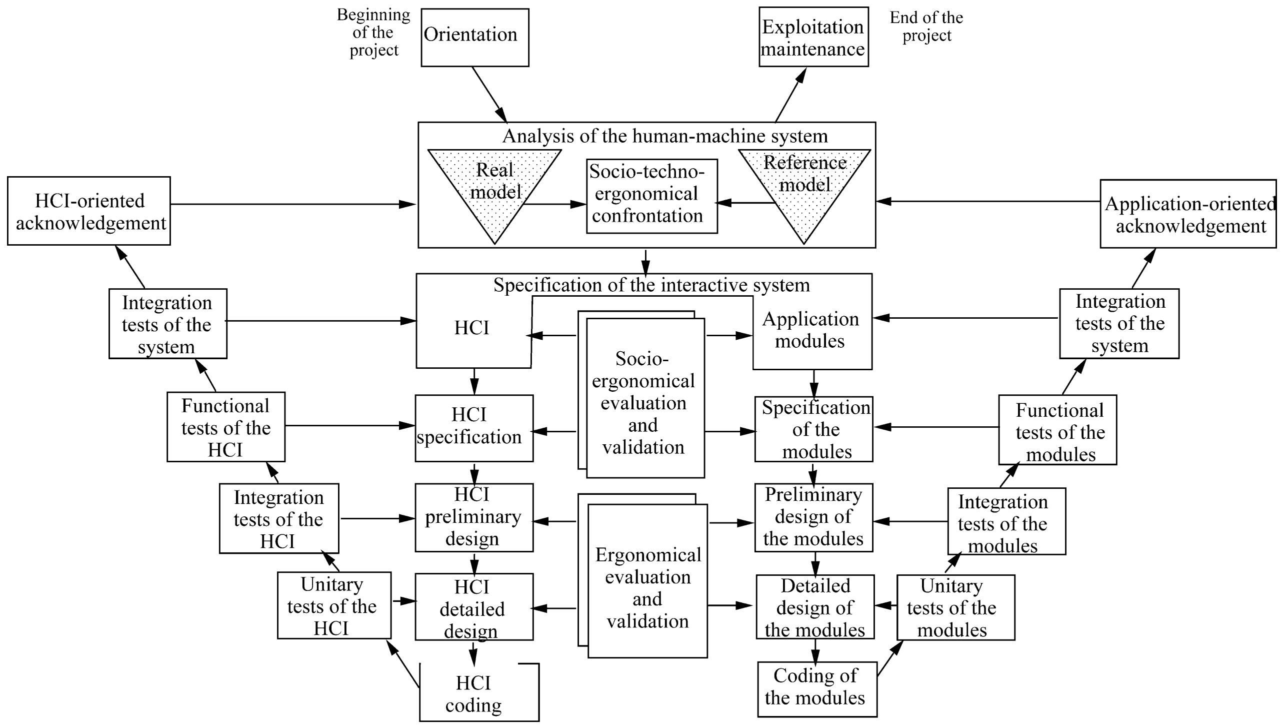
the various software engineering stages necessary for the development of an interactive system, and at the same time differentiates the actual interface (left-hand part of the model) from the support (or applicative) modules which may be accessed from them (the right-hand part) (Fig. 6). Nabla is based on a progressive confrontation between a real model and a reference model, in which the reference model corresponds to the so-called ideal human-machine system, considering the points of view and the needs of the various users concerned by the human-machine system in question. The result of this confrontation leads to the identification of relevant data in order to specify an interactive system which is adapted to the informational needs of the users, as well as to the needs regarding the user-support module cooperation mode. The specifications are then evaluated and validated from a socio-ergonomic point of view, in order to check the relevance of the integration of new solutions into the human-machine system in question. The evaluation aspect is situated at the centre of the project, suggesting an iterative process in the left-hand parts as well as in those in the right-hand parts. It ends with an acknowledgement stage which is symbolically separated into an HCI-oriented acknowledgement and an application-oriented module acknowledgement.

## 2.3 A discussion concerning these models

The HCI-enriched models we have presented prove to be unequal, with a varying degree of closeness to software engineering. However, they all include promising ideas as regards the problem posed by human-machine systems.

Nevertheless, a certain number of limits can be mentioned; for example, in the case of the star model, the task analysis stage is only indirectly validated by a prototype. Indeed, the prototype only concerns a part of the development of the interactive application, i.e., the contractual, external specifications of the application. As regards the Hartson and Boehm-Davis model, the authors suggest that once the presentation has been confirmed, the software development process should then take place for the functional aspects of the application according to the classic software engineering methods. Consequently, any implication of the task

**Fig. 6** A Nabla model for the development of interactive systems


Beginning of the project
Orientation
Exploitation maintenance
End of the project
Analysis of the human-machine system
Real model
Socio-techno-ergonomical confrontation
Reference model
HCI-oriented acknowledgement
Application-oriented acknowledgement
Specification of the interactive system
HCI
Application modules
Integration tests of the system
Integration tests of the system
Socio-ergonomical evaluation and validation
Functional tests of the HCI
HCI specification
Specification of the modules
Functional tests of the modules
Integration tests of the HCI
HCI preliminary design
Ergonomical evaluation and validation
Preliminary design of the modules
Integration tests of the modules
Unitary tests of the HCI
HCI detailed design
Detailed design of the modules
Unitary tests of the modules
HCI coding
Coding of the modules

in the functional part is excluded, unlike other methodological design frameworks such as MUSE*/JSD (Lim and Long 1994), TRIDENT (Bodart et al. 1995), GLADIS++ (Buisine 1999), etc. This limitation is also found in the Nabla method. Indeed, the Nabla method does not clearly explain the modelling of the user and the human tasks by showing their connection with the interface specification (they are, in fact, integrated into the analysis box of the human-machine system). Moreover, like the V model, Nabla expresses itself in a series of very limited returns, which can be a handicap for iterative design. The model does not indicate anything concerning the making of a prototype (this idea only appears in its original, literal description). On the other hand, the Nabla model is an interesting attempt on the part of software engineering to connect to cognitive ergonomics, done here by taking human factors into account, and also through ergonomic assessment.

Globally, we can note here that most of the models (whether they are enriched or not) do not suggest formal use and user models within the process. This is a pity, given the active research currently being performed on this subject.

As regards evaluation, several authors suggest that it could be of two types: formative (during the design and the development of the interactive system) or summative (after a full system has been deployed) (Hix and Hartson 1993; Hix 1995). In the models described above, it could be said that, in the waterfall and V models, along with the Curtis and Hefley model, the assessment is potentially summative; in the spiral, Hartson and Boehm-Davis, and Hix and Hartson models, the assessment is more formative; the nabla model could potentially integrate the two types of assessment (formative by comparing the two models and working with a socio-ergonomic point of view, summative in the acknowledgement stage).

Therefore, it can be said that no perfect model exists; they all have their strong and weak points.

## 3 A U-model for the design and evaluation of an interactive system

In the previous section, we have highlighted limits inherent in the development models. This is the context in which, for the past few years, our research projects have been aimed at defining a theoretical and methodological framework for the design and evaluation of interactive systems. This framework is based on a process, called the U-model (Fig. 7). One of the striking characteristics of this model is that it situates stages—which do not exist in classic software engineering models, which remain very general—during which human factors must be considered by the development team.

The U-model is structured into two phases, as can be seen in Fig. 7: (i) a descending phase with the modelling of the human-machine system, which leads to its implementation (ii) an ascending phase made up of the evaluation of the overall system, according to system efficiency criteria and also strictly human criteria.

It should be noted that this model has been partially or completely applied in many industrial projects over the past ten years. On the practical level, we have introduced several levels of description corresponding to the different stages of the U-model's development cycle. Each description level gives rise to a model which, through successive transformations, will guarantee a continuity of analysis throughout the project.

### 3.1 The U-model descending phase of design and creation

The beginning of this phase starts with two essential steps which take place simultaneously and which mark the beginning of the project: (i) the analysis of what exists and what is needed (ii) the analysis of the process and its environment.

The analysis of what exists is intended to provide a structuring framework, as regards future activities as well as technical solutions. In the applications we are concerned with, we want to design new systems based on operational systems or others, which correspond to new tasks or to tasks which are the result of integrating several existing tasks which have been performed separately up to now. Based on the analysis of the activity, the main aim of the analysis is to clarify the user's knowledge of the task along with the representation he/she has of it (Bainbridge 1978; Hoc and Samurcay 1992). At this level, the task description must be free from the constraints of existing tools which are imposed and for which the user develops compensatory strategies in order to resolve any possible weaknesses (Reason 1988). The analysis activity can be performed using various techniques: interviews, written work reports, expert analysis reports, questionnaires, critical incidents, monitoring, etc. (De Keyser et al. 1987; Wilson and Corlett 1996). This analysis can also obtain information from written procedures and from experts in the field.

The analysis of needs is not only concerned with a factual view of the existing system, but also with the underlying need which is expressed through what exists and also through wishes voiced by the users or assessed by the ergonomists. The specification of needs should therefore be able to tackle what exists through the organisation and work requirements of the users; it should specify and formalise their needs and especially a set of requirements as regards the future interactive system.

All of the data resulting from this phase must be transposed, if possible, into a single "source" model. This model, along with the representation support it uses, is the break point in the project. As such, it must act as a stable framework for the development of the

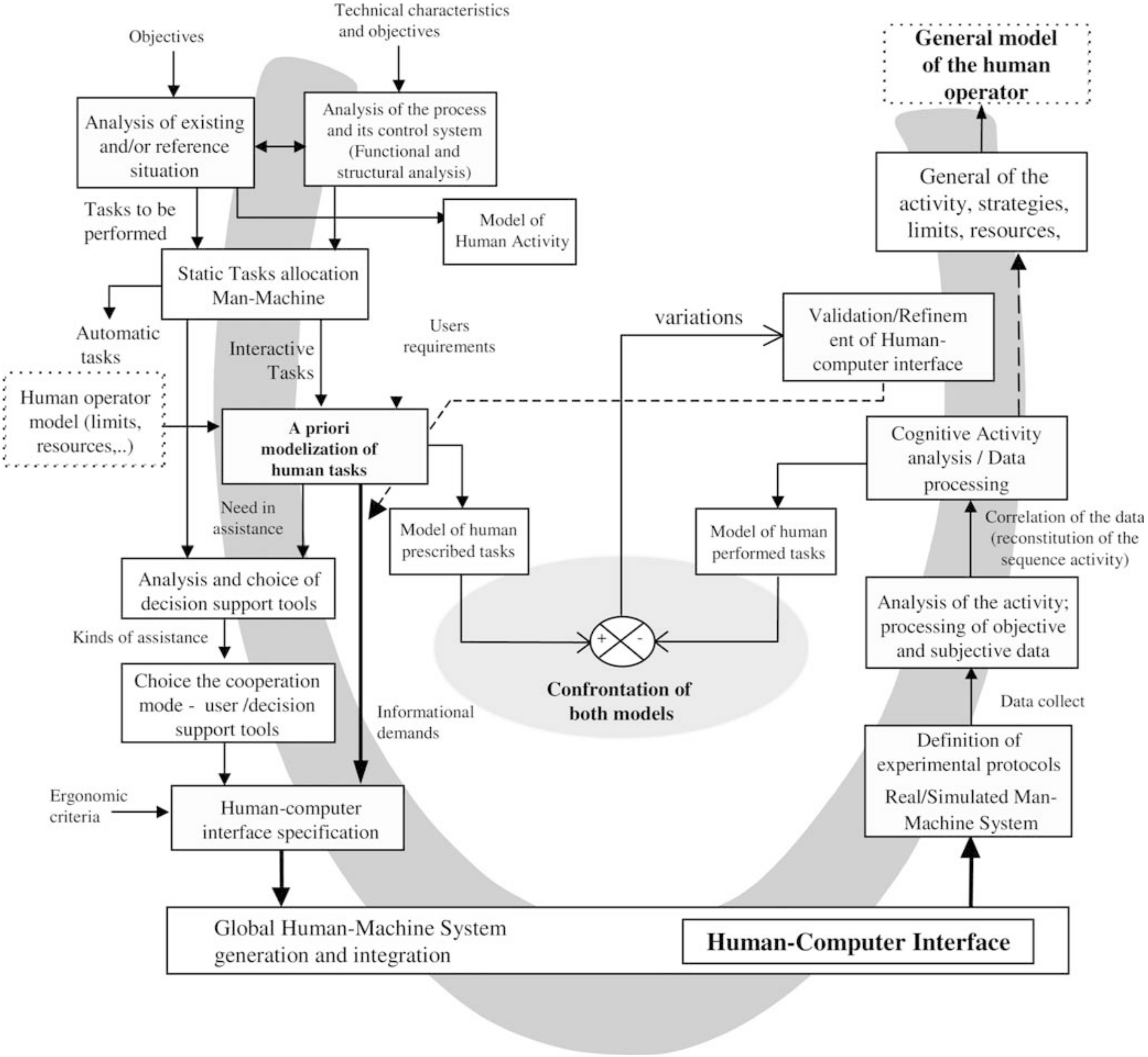

Objectives
Technical characteristics and objectives
General model of the human operator
Analysis of existing and/or reference situation
Analysis of the process and its control system (Functional and structural analysis)
Tasks to be performed
Model of Human Activity
General of the activity, strategies, limits, resources,
Static Tasks allocation Man-Machine
Automatic tasks
Interactive Tasks
Users requirements
variations
Validation/Refinement of Human-computer interface
Human operator model (limits, resources,..)
A priori modelization of human tasks
Cognitive Activity analysis / Data processing
Need in assistance
Model of human prescribed tasks
Model of human performed tasks
Correlation of the data (reconstitution of the sequence activity)
Analysis and choice of decision support tools
Analysis of the activity; processing of objective and subjective data
Kinds of assistance
+
-
Confrontation of both models
Choice the cooperation mode - user /decision support tools
Informational demands
Data collect
Definition of experimental protocols
Real/Simulated Man-Machine System
Ergonomic criteria
Human-computer interface specification
Global Human-Machine System generation and integration
Human-Computer Interface

**Fig. 7** The U-model (Abed 1990, Millot and Roussillon 1991)

following stages, especially by providing the design teams with a starting point:

- To define the global data structure for the system (the information handled) and the actions each user is likely to perform using the human-computer interface
- To identify the main functions of the system
- To trace back all the ergonomic constraints linked to the operator's mission and his/her work context in terms of needs
- To identify the division of tasks between the operator and the machine (cf. hereafter)

In parallel, the analysis of the process makes it possible to list the technical constraints according to the various foreseeable execution modes. The definition of a process model can be based on methods which enable a better approach not only to the functioning of the system and the sub-systems which compose it, but also to any foreseeable dysfunction. Two main types of method can be identified (Fadier 1990; Villemeur 1992).

The first methods, which are generally well-known to automation and computer scientists, are intended for the analysis of a normally functioning system and its description according to structural and functional aspects. As examples, we can mention SADT (Marca and McGowan 1988), SA-RT (Hatley and Pirbhai 1991), SA (DeMarco 1979), MFM (Lind 1990), the object-oriented methods, especially UML (Booch 1994) and Petri nets (David and Alla, 1994).

Other methods, mainly stemming from the fields of system maintenance and reliability, can be used as a complement to the analysis of a normally functioning system, such as FMECA (Recht 1966) and FTA (Hassl 1965), for example. The use of such methods aims at defining the various foreseeable cases of dysfunction and at determining the reparatory actions to be taken into account in its composition. These actions also lead to the definition of the prescribed human tasks in the human-machine system.

After these two preliminary stages, a model of the human-machine system can be created in order to identify and organise all of the tasks to be fulfilled by the operator-machine couple. Several existing models deal with task modelling, based on the principle of hierarchic decomposition. This principle makes it possible to gradually introduce levels of detail which are increasingly fine (the breakdown of tasks into sub-tasks) according to the structure of the system to be created (Buisine 1999), or "Concur Task Tree" models (Paterno 2000). The representation formalisms of these models make it possible to note the properties (attributes) of each task and the way they relate, thus expressing the dynamics of the model, i.e., the logical and/or temporal constraints. It is also important to represent the data in connection with the tasks in order to link the application data to the treatment it authorises.

Following the modelling of the human-machine system, a distribution[1] of tasks between the machine and the human operators can be performed in two ways, either in parallel or after the decomposition process, answering the question "who does what?". Tasks are distributed in relation to the characteristics and treatment capacities of each one, as there are no strict conditions to be respected. Amongst the influencing factors, the following factors can be mentioned: the repetitive elements, the memorisation capacity, decision taking, errors and error correction, rapidity of treatment, etc. It should be noted, however, that once the distribution has been performed, the human-machine system is definitively rigidified. Each task in the interactive system has a degree of interactivity. Three main categories of task can therefore be identified : (i) tasks in which the user alone is implied, called a manual task (ii) tasks in which the applicative aspect alone is represented, called system tasks (iii) tasks involving varying degrees of collaboration between the user and the system, called interactive tasks.

Following the allocation of tasks, the method consists in concentrating on the tasks in which the human user appears as an actor, i.e., the interactive tasks. It is necessary to be able to specify the interface for each task, in particular as regards the information to be displayed and the technical reactions of the system, using the probable behaviour of the operator as a basis. It is a question of analysing and modelling the behavioural aspect of the human-machine interaction in interactive tasks, according to the goals to be achieved; this is the next step in the descending phase. At this level of task modelling, the aim is to establish the *prescribed* activities the users will have to perform. This must take into account the model of the various users in terms of limits and physical and cognitive resources, relating not only to the acquisition and processing of information, but also to the existing activity models of the phase in which the needs and existing elements are analysed. The user model is the subject of a vast and complicated field of research aimed at understanding the human reasoning process. In this field, a number of methods and models based on a theory (or elements of a theory) of human interaction with machines and environment have been proposed: action-related theories (Norman 1986; Tijus and Poiternaud 1996; Theureau and Jeffroy 1994), activity theory (Nardi 1995; Fréjus 1999; Wehner et al. 2000), problem resolution models and human error models (Rasmussen 1986; Cacciabue et al. 1992; Moray 1997; Amalberti 1997). The latter models try to understand how and why human errors appear. Other research projects into artificial intelligence attempt to model concepts of the generation and the integration of agent plans, along with intentional states such as belief and intention (Rubin et al. 1988).

The modelling of interactive tasks will refer to procedures made up of elementary operations which the operator is supposed to perform to carry out the task. These procedures formalise the dialogue sequences defining the strategies and requests of an operator which are necessary to achieve the set goal. Each elementary operation has one goal which is expressed in the name of the operation associated to a "domain object", for example, "modify *speed* of a train". An object can be a single unit, or made up of several objects of the domain. There are two types of elementary operation: physical and cognitive. The physical elementary operation is expressed by visible actions on the human-machine interface, such as the input of a chain of characters, the selection of a value, etc. On the other hand, the cognitive elementary operation cannot be seen and represents mental activity such as a comparison, a choice, a decision or a combination of these three activity units.

The detailed specification of the action procedures must refer to two types of complementary analyses:

- *In terms of planning* which concerns the detection of rules or heuristics to be used and decision strategies to be taken into account. For example, in railway control, the detection of conflicts brings different types of reasoning into play which can be used by the controller, such as the intersection at kilometric points, two-way traffic, use of depot lines, etc.
- *In terms of optimisation* of the choice of action which defines a set of criteria for the development of

[1] The distribution of tasks is one of the points overlooked by current methods of task modelling. The division brings an element of a solution to establishing the link between the analysis and design stages, by identifying the agent or agents which will have the responsibility of executing the task in question (Balbo 1994; Buisine 1999; Paterno et al. 1997)

possible action paths leading to the performance of the task. These criteria include notions of risk, safety, efficiency, feasibility, cost, etc. For example, in railway control, a situation of interference between trains calls upon a conflict resolution strategy based on criteria such as spacing, timing, itinerary, speed or a combination of these criteria. The criteria directly affects the performance of the trains and is also applicable to the tasks of regulation and situation takeover.

In this way, the counting of the solutions possible in terms of planning and optimisation should show up the informational needs of the users, corresponding to the data necessary to perform the different tasks, as well as to their needs in support tools (functions), which can be in the form of decision support systems (alarm filtering, diagnosis, planning, etc.).

For task modelling, researchers can use propositions coming from both software engineering and the cognitive sciences. The orientation of the cognitive sciences is directed towards the way of characterising and identifying tasks, thus contributing to the task analysis phase, for example, TKS (Johnson et al. 1991; Johnson 1999), MAD (Sebillote 1995; Scapin and Pierret-Golbreich 1990) or GTA (Van-eylen et al. 1996). On the other hand, the orientation in software engineering is directed more towards the provision of notations for the representation of tasks and their relationships. It should be noted that today the formalisms lack formal engineering techniques for task modelling. The expression of task models in an informal manner leads to the risk of incorrect interpretation by the people involved in development. In fact, informal modelling does not encourage the building of reliable systems, given that it does not eliminate the possibility of the incorrect interpretations of models. However, from another point of view, the techniques should provide notations with sufficient power of expression to be able to describe possible actions clearly, so that they are not too complex in order to be usable by people who have a limited knowledge of mathematics. In this context, the support tools for the modelling and analysis of tasks provide valuable help for the building of task models and their use by designers; concerning this, see the following software environments: PetShop (Navarre et al. 2002), GLADIS + + (Buisine 1999), MAD* (Gamboa-Rodriguez and Scapin 1997), TAMOT (Lu et al. 1999) and E-TOOD (Abed 2001; Tabary 2001).

The task model resulting from this stage is the specification source for the human-machine interfaces and support tools, and also contributes towards other goals such as: the predicative analysis and evaluation concerning system usability, the discussion support between the various actors in the project and the reference model for the analysis of the real activity, as shown in Fig. 7.

A preliminary evaluation of the task model can be performed at this level. This evaluation is intended to check whether the system model and the task model are compatible. The verification consists of checking whether the task model is included in the system model, which proves that the user can perform his task with the system as it is defined in the model. Whenever the result is deemed unsatisfactory, a modification is introduced at the system model level in such a way as to produce task models which are compatible with the model of the new system.

Once the informational needs and support needs have been identified, it then becomes possible to define and specify an architecture for the human-machine interface. Its specification aims at analysing and defining the behaviour of the interface. It is different from the specification activity in classic software engineering in that the interactions described in the specification are concentrated on the relationships between the user and the interactive system. It is a question of strictly identifying the ergonomic needs and techniques, and then defining the number of screens to use, the display sequences, the information presentation modes, the activation modes for the various support tools, the modalities for human-machine dialogue, etc. This passage must also comply, in principle, with the temporal and structural relationships of the task model produced beforehand. The dynamics of the dialogue become more difficult to describe when the user is given a maximum degree of freedom; he can then trigger several dialogue lines at the same time (a multi-line dialogue). These constraints make it necessary to specify the behaviour of the human-machine interaction both coherently and with no ambiguity. In order to overcome these constraints, the specification can be made easier by the joint use of a set of techniques.

The formalisms for the specification of interactive systems currently available are numerous and varied. All of them have advantages and disadvantages, and no single one of them can be regarded as an exhaustive specification. In the work of Brun (1998) and Jambon et al. (2001) we find the criteria[2] for choice allowing the evaluation of the respective qualities of the formalisms and the method of choice[3] of a formalism. The choice of a formalism or of a specification notation is a strategic decision during the development process of an interactive system. A badly adapted formalism will make the specification activity difficult and increase the specification time whilst discouraging the development team; at worst, it will be the source of inaccuracies which will lead to design errors. There are several classifications of these formalisms, for example: graphic or textual (Dix et al. 1998), with states and events (Tarby and Barthet 1996), according to a user or system perspective (Harrison and Duke 1994) and according to the origin of formalisms (cognitive sciences, graph theory and algebraic approaches) (Brun 1998).

[2] This study uses twelve criteria, grouped together into three categories: (1) power of expression (2) generating capacities (3) extendibility and usability.

[3] These authors introduce four sequential stages in the choice of a formalism: (1) choice according to the activity (2) choice according to the appropriateness of models (3) choice according to the power of expression (4) choice according to the degree of usability.

It is very important during this stage to take into account a set of criteria resulting from software ergonomics, relating for example to the coding of information, to coherency, to readability, to the various representation modes possible, etc., whilst aiming to avoid as far as possible the sources of human error coming from problems of perception, identification or uses of information, for example. For this, it is possible to turn to recommendation manuals (Smith and Mosier 1986; Vanderdonckt 1994), as well as to style guides (Windows, MacIntosh, OpenLook, Motif...). However, the use of the style guides will be reinforced advantageously by the presence of a specialist in human-machine communications. It should be noted that the specification must also conform to the norms and/or standards applicable in the application field.

The specification must lead to the development of two model types: (i) an abstract interface model which defines the information to be presented to the user in an abstract manner, as well as the dialogues allowed to interact with this information in terms of abstract interaction objects (ii) a concrete interface model which specifies the return of this information in terms of concrete interface objects, corresponding to elements of the tool box (menus, check boxes, etc.). The specification of the human-machine interfaces leads to the last stage in the descending phase of the U-model, that of the creation and integration of the complete human-machine system or of its prototype on site and/or in a simulation situation. This implementation stage transforms the concrete interface specifications into a representation which can be used directly by a graphic tool box or a human-machine interface generator. There are three types of tools which can be used: (i) generators of source codes in a given language (ii) UIMS type generators (iii) interpreters which do not generate an implementation file but which interpret the model directly during execution (Myers 1993; Fekete and Girard 2001).

It should be noted that over the past ten years, a new research orientation has been emerging, based on the paradigm of model-based user interface design (MBD). This research movement aims at federating tools, formalisms and methods, with a view to creating units, which are grouped together in the development environments and more or less cover the development cycle. The major disadvantage of MBD type approaches is the complexity of the models and notations which are generally difficult to approach and manipulate (Myers 1995). They are therefore generally equipped so as to encourage the understanding of their complexity. The term generally used to refer to these environments is model based interface development environments (MB-IDEs) (Szekely 1996).

## 3.2 The U-model ascending evaluation phase

As current knowledge concerning the human operator and the cognitive aspects linked to the work place is too incomplete to be able to envisage an open loop design, an evaluation stage must be used. This is the role of the ascending phase of the process.

The evaluation of a human-machine system consists in checking that the operator is capable of performing his or her task using the interface provided. If this interface has not been designed correctly, it can lead to the rejection of the system. On the other hand, a well-designed interface will make possible a harmonious integration into the operator's task of the capacities of the system which has been developed, by providing the operator with precious help and support. Between these two cases, the consequences on the user's work can be varied (Kolski 1997). Two properties are usually explored in the evaluation of a human-machine interface: usefulness[4] and usability[5] (Shackel 1991; Grudin 1992; Farenc et al. 1996, Bastien and Scapin 2001). Many authors have given their own definition of these properties, or have characterised their attributes so as to be able to measure them (Senach 1990; Nielsen 1993; Grislin and Kolski 1997).

This field of research is currently booming to such an extent that many methods are available and several classifications of these methods have been suggested. As quoted by Grislin and Kolski (1997), a distinction is often found between predicative approaches and experimental approaches; this is the case, for example, in the classifications developed by Nielsen and Molich (1990) and Hix (1995). The predicative approach is performed on a theoretical representation of the system and requires neither a real system nor a real user. On the other hand, the experimental approach is based on a real system (a mockup, a prototype, etc.).

The U-model recommends an approach based on the diagnosis of use. This approach is applied when there is an experience in using the overall system, or part of it. In this evaluation, we generally concentrate on the performance of the entire system, on the one hand, according to user behaviour during interaction with the system (for example the time required to perform a task, the accuracy of the result, the number and type of errors, the difficulties encountered, compliance with the installation's safety recommendations, the operator's opinion, especially concerning the dialogue interface and any support systems and finally the operator's work load), and on the other hand, according to the system in terms of differences between the production and the aims.

[4] Usefulness concerns the appropriateness which exists between the functions supplied by the system and those necessary for the user to perform the tasks assigned to him correctly. It is linked more to the functionalities of the software.

[5] Usability takes into account the quality of the man-machine interaction in terms of ease of learning and use, as well as documentation (Grislin and Kolski 1997). Generally speaking, the addition to the human-machine interface of functionalities intended to make for easier use sometimes has the opposite effect of complicating the user's task (Palanque 1997). Scapin and Bastien (1997) prefer to use the term "ergonomic quality" of interactive software to show that as much interest is given to the functionalities of the software as to its interface.

As shown in Fig. 7, the ascending phase requires the definition of strict experimental protocols, intended to define not only the way in which the tests are performed, but also the data to be obtained (Millot and Debernard 1993). This stage is not simple in that some data cannot be directly observed and does not make it possible to measure the difficulties encountered by the users. Some measurements are manual, whereas others concern non-verbal behaviour, thus requiring equipment (such as a monitoring system, a measurement of heart beats, a measurement of eye movements, etc.). It also requires the choice of representative users performing representative tasks in a representative context (McKenna 1996; McGee et al. 1998). The measurements can be performed very early on in the design process; that is to say, when the design choices have merely been envisaged, unlike the design test approach.

The cognitive analysis of activities and the processing of the resulting data can be structured in operational sequences and formalised into models of the task performed or the real task, and can lead to a comparison between the tasks truly performed by users and the prescribed tasks defined in the descending phase (Fig. 7) (Abed and Angué 1994).

The principle of modelling the operational sequences consists in comparing on the one hand the eye focus sequences and on the other hand the objective data selected by the observer model, i.e., the information displayed and its content, the operator's physical actions and the machine events. This objective data is then enriched and completed with the functional inter-operator dialogues (if there are any) and the comments of the operators on their own activity, and with questionnaires and individual self-confrontation interviews with the operators after each manipulation session. The questionnaires provide information concerning the attitude and opinion of the human operator as regards the human-machine interaction. On the other hand, the self-confrontation interviews make it possible to obtain complementary explanations concerning the operator's cognitive behaviour, to confirm (or otherwise) hypotheses made by the analysts, and so on (Theureau 1992). The correlation represents a flow of data on observable activity which, when broken down and analysed, makes it possible to reconstruct the operator's behaviour and his or her real task. Thus, the confrontation between the real/prescribed task leads to the identification of the mental processes brought into play as well as the resulting work rules, and also to obtaining a general behaviour model which groups together all the strategies used by the various operators to perform one task. The principle of the confrontation mechanism[6] consists in generating a reference model (initially corresponding to the prescribed task model) and in enriching it iteratively using the differences noted between the prescribed task model and the models of real activity. The reference model obtained is considered to be exhaustive once all the activity sequences concerning one task have been confronted. It is called the "general model" in Fig. 7. In this way, it is possible to obtain an exhaustive description of the strategies used by the operators in order to perform any given task.

The result of the confrontation makes it possible either to validate the human-machine system or to show up its shortcomings and to improve it progressively, especially as regards the human-machine interfaces and support tools. The final model resulting from the confrontation thus makes it possible to generalise specific human behaviour in particular work conditions, which can be used again in situations with similar systems.

## 3.3 Conclusions on the U-model

Our general model enables us to better situate a group of notions which are essential (from the human factor angle) for the development of interactive systems and which do not appear clearly from classic software engineering cycles (such as the waterfall, V or spiral models, etc.).

In its original version, this model made it possible at the time to begin to position the first stages which appeared to be fundamental (the shaded stages on Fig. 7) as regards the design and assessment of interactive systems (Abed 1990; Abed and Angué 1990; Millot and Roussillon 1991). Since then, over the past ten years, it has been progressively enriched by adding stages, most of which are the result of research carried out during numerous industrial projects (air traffic control (Abed 1990; Millot and Debernard 1993), railway supervision (Ezzedine and Abed 1997), chemical process supervision (Kolski et al. 2000), etc.), for example:

- The stage called "Analysis of existing and/or reference situation" did not exist previously
- The same is true of the stage called "Analysis and choice of decision support tools"
- The comparison or confrontation of the activity model to a reference model has also become increasingly detailed according to the experience acquired
- Other stages are currently being validated concerning the integration and use of ergonomic and expert knowledge in the initial phases of the development process (Abed 2001); some of these will be presented in a later section.

The model can be adapted according to the specifications of the application. Thus, the case study presented in the next section of the paper explains how it was necessary to adapt it in the framework of the design and evaluation of an interactive decision support system in the field of rail transport.

[6] Further information concerning the methods of confrontation using algorithms based on the Petri net theory is to be found in Bernard (1994), Palanque (1997) and Abed (2001).

## 4 Case study: the INFRAFER project

### 4.1 The industrial context

The case study involves a joint project between CORYS TESS, RFF (Réseau Ferré de France) and the LAMIH, performed as part of a project named PREDIT (1999–2001) which was sponsored by the French Ministry for Education, Research and Technology (Paulhac et al. 2001). This work resulted in an interactive decision support system named INFRAFER (INFRAstructure FERroviaire in French) which was intended to help the company RFF, the owner and manager of the French railway network, to manage its investments in its infrastructures. Insufficiency in infrastructure may only be revealed by indicators such as the rail transport capacity. This capacity is expressed according to the journey time; it is defined by "the number of trains with a given journey time which can travel on a section of track".

The aim of the system is therefore to find the capacity of the existing infrastructure or of a fictive infrastructure. The comparison of these two types of infrastructure will account for the true economic factor of an investment (which can amount to several million euros).

### 4.2 An analysis of the human-machine system and the existing systems

As we have explained previously, the analysis of the human-machine system is an important part of the model. A system must be designed to satisfy the end users to the greatest degree possible; for this, it must take user needs into account. The analysis also serves to study the strong and weak points of the existing systems.

The needs of the users (in our case, the experts from RFF) were established following many meetings with the company, including the examination of case studies on rail transport capacity. The aim of the system was defined with the company's experts: it must indicate the number of additional rail convoys on a line in one direction and must give the times of departure and arrival according to parameters given by the user: the case study schedule interval, the spacing time between trains on departure and arrival, the type of rail convoys to be inserted, etc. The system must also be user-friendly, and simplify data entry; it must be simple to implement and also present its results as clearly as possible so that non-experts are able to understand them.

The systems and methods which exist in the field of rail transport were studied and evaluated as regards the needs of the users. It was found that the methods of evaluation of saturation and capacity of railway infrastructures are many and varied; they can be classified according to the following approaches.

The analytic formulae are methods based on the evaluation of average minimum times of successions $T_s$ of the various trains. These formulae are different from each other in their different methods of approaching $T_s$ and by the different margins adopted according to the level of quality required. Included among them is the UIC form developed in 1979 (UIC 1979), the CFF method (from the Swiss Federal Railway Company), the formula used by the SIMON software programme, the NS and FS formulae (quoted by Hachemane 1997). The UIC formula is mainly based on the average length of succession; it does not give the capacity for a given type of train. The CFF formula gives the rate of saturation and not the capacity. The method used by the SIMON programme is applicable mainly to cases of homogenous traffic. The FS formula uses hypotheses which are too simplistic, whereas the NS formula uses data which is difficult to obtain. The use of these methods does not provide a precise result concerning capacity.

The probability formulae are methods which can be used when the exact schedule grid is not known. They are, therefore, based on a probability evaluation of the distribution of trains and they form hypotheses on the distribution of traffic. The DB and Schannhäusser formulae, quoted by Hachemane (1997) are included in the probability methods; they both use the hypothesis that the distribution of the number of trains appearing in a given time period is governed by a Poisson law. The method developed by Florio is based on the probability of two trains being in conflict (Florio et al. 1998; 1995). This method is not adapted to our situation because the users have schedule grids and they are seeking the exact number of rail convoys which can be added.

The schedule construction methods are methods which start from a given schedule grid and use theories to develop the densest grid possible with no "convoy loss": this therefore corresponds to the most saturated situation possible. It has two variants: the compacting method which, using a traffic graph corresponding to the line section being studied, has as its principle to narrow the gaps between convoys as much as possible without changing the ordering, the journey time and the immobility time. A "compacted time" is thus determined and the time available can be deduced through the difference with the reference time. This method is used by the Computer Aided Timetable Design systems used by the SNCF, called SOFTIME by SYSTRA (SOFRERAIL 1992). The compacting method makes it possible to know the quality of a schedule grid more than the availability of convoy space. The CAPRES system (a support system for the analysis of capacity of railway networks) consists in developing the most saturated schedule on the network. This system has several disadvantages, the main one being that the user must enter the entire infrastructure (Hachemane 1997).

The simulation methods are computer methods which do not perform theoretical calculations but which simulate the traffic of various known trains and the various events which can happen on a network. It is therefore possible to have a visual idea of the level of quality and strength of a grid. These methods mainly allow the verification of feasibility of a given schedule grid. There are a number of software programmes based on the same principle, such as FASTA (ESA 1991), RAILSIM (SYSTRA Consulting) and SISYFE (Fontaine and Gauyacq 2001).

The comparison of all these methods in relation to the needs of the user shows that up to now, there is no method which completely satisfies the criteria sought by the ultimate users. The methods presented are either complicated to use, or they do not give the exact number of additional convoys. The U-model presented previously will therefore be adapted for the needs of the project.

## 4.3 The design and creation of an interactive decision support system

The existing formulae and methods for the calculation of railway capacity are not appropriate for the needs of the users (who, in our case, are also experts in their field). As the financial consequences of the decisions taken are enormous, the need for correct results which satisfy the users led to the enrichment of the U-model by a knowledge extraction phase (Fig. 8). This phase makes it possible to establish a method for the calculation of capacity following the method used by the experts.

The knowledge extraction phase is broken down into several stages. The first stage enables the acquisition of a degree of competence for dialogue with the experts in the field. It must begin with a bibliographical acquisition of knowledge. Similar systems which have been developed in industry or in laboratories must also be studied, thus enabling the designers to become familiar with the calculation principles and the presentation of information linked to the field in question.

The second stage involves the acquisition of knowledge from experts during regular meetings, using interview techniques (Olson and Rueter 1987; Preece et al. 1994; Macaulay 1996) and the analysis of written reports and documents (Sperandio 1991; Maguire 2001), along with case studies. With an aim to illustrating our point using a concrete example related to our case study, Figs. 9 and 10 show respectively (1) an extract from the written reports of an expert during a meeting to explain the notions which appeared to be essential to him, and (2) an extract from the documents used by the experts as a basis. The report extract has several parts: a diagram showing the functioning of a railway device (an automatic luminous block), the rules for the construction of a graph which both the expert and the system must respect, a commentary which situates the context of the diagram and the resulting formulae.

This disparate knowledge is broken down into concepts and elementary rules. A global model of the activities is made (an extract is shown in Fig. 11; it is based on the system analysis design technique (SADT) method). The analysis of the railway infrastructure includes several stages; first, the data necessary for the analysis must be obtained, followed by the action required which includes either the increase of capacity (through the calculation of residual capacity) or the evaluation of the schedule grid in relation to the infrastructures.

►

**Fig. 8** An enriched U-model (Lepreux et al. 2001)

Finally, the third stage consists of an analysis and evaluation of the concepts and rules by experts until an agreement is reached. When this stage is performed, great importance is given to the way in which the experts imagine an interaction with the system as well as the presentation of information. At this stage, models on paper and then software models are presented to the future users of the decision support system (who in our case are expert users); following an iterative process of assessment, explanation, modification and validation (Lichter et al. 1994), we aim to finish up with a version which corresponds as closely as possible to the needs (cf. Fig. 12). The copy of a screen at the bottom on the left shows the final window of the system; it demonstrates that this window is the result of the modelling of expert knowledge and is true to their work approach.

### *4.3.1 The architecture of the interactive decision support system*

Using the analysis of expert knowledge previously performed, the essential functions linked to the expert approaches were identified. They were structured according to a set of modules. Figure 13 represents the modules integrated in the software architecture (by circles) and their relationship with the data files (rectangles). An arrow directed from a module towards a file indicates that the module is writing in the file, whereas an arrow directed from a file towards a module indicates that the module is reading the file (Lepreux et al. 2001). The system's architecture is based around eight modules.

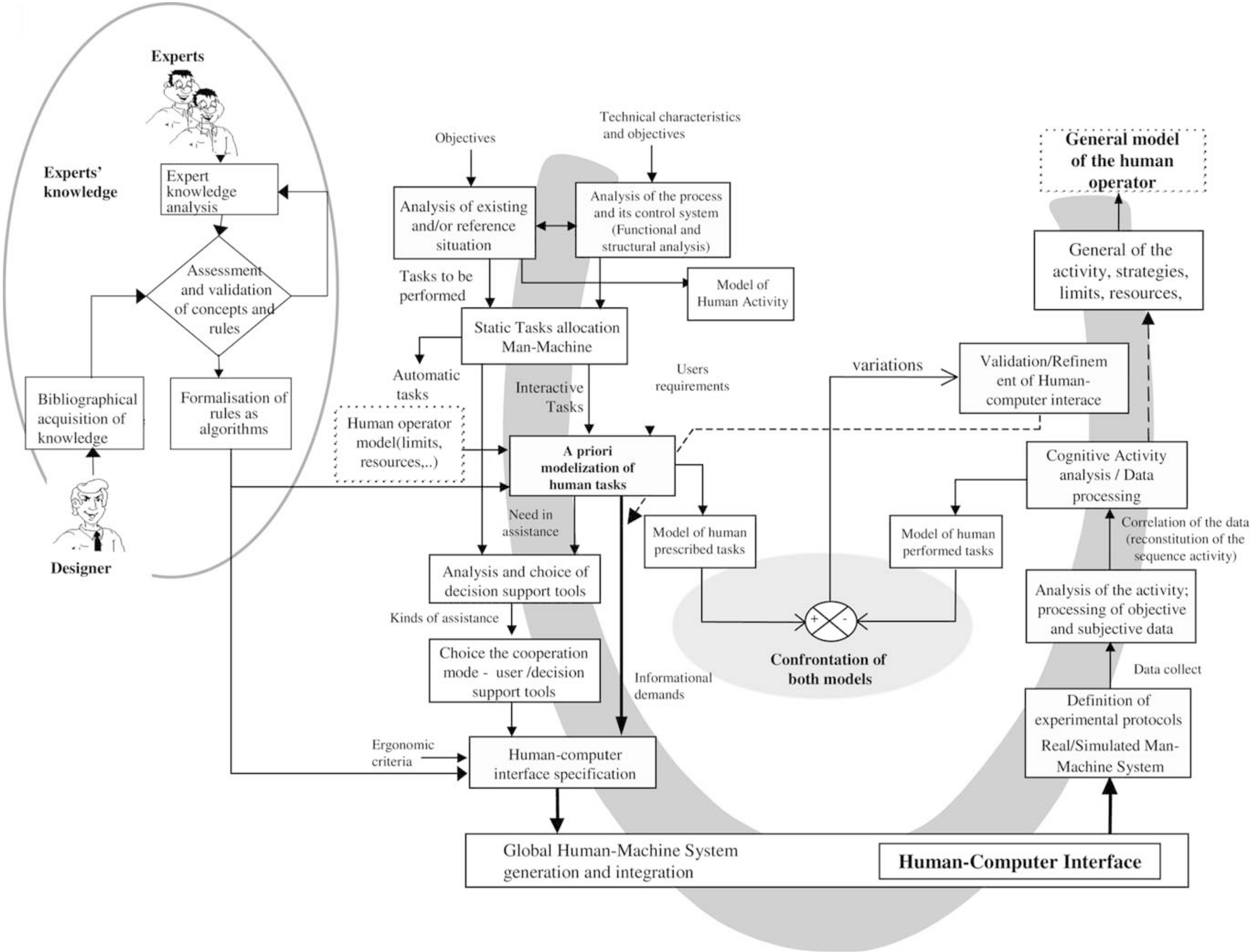
The *first* module directs the other modules. The *reference mark editor* is the module with which the user creates, modifies or deletes reference marks. The role of the reference marks is to situate all the units representing the railway infrastructure.

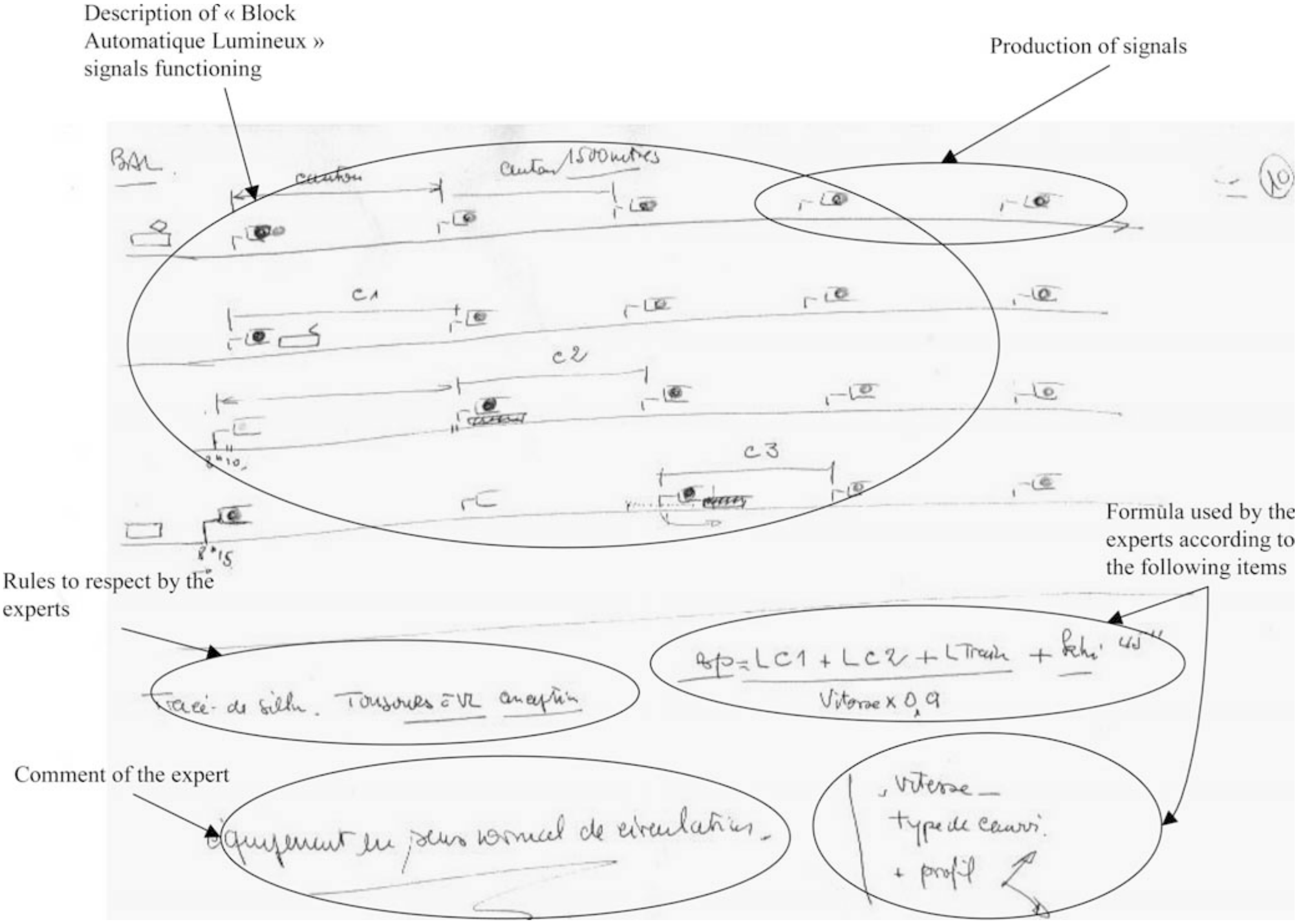
The *train editor* enables the user to create, modify or delete trains and all their physical characteristics (composition, locomotives...).

The *section editor* gives the user the possibility of creating, modifying or deleting sections which describe all of the elements of the infrastructure. Several exploitation scenarios can be built and tested in the same section.

The *scenario editor* enables the user to create, modify or delete scenarios. The scenarios cover the typical running of all trains in a basic traffic pattern or in a modified traffic pattern.

The *calculation module* has the role of determining the residual capacity of a typical running on a portion (one or several segments) and/or a complete route in a time interval chosen by the user. The result of the various possibilities for the placement of supplementary convoys is then used by the simulation module to confirm or invalidate the schedules chosen.


Experts
Experts' knowledge
Expert knowledge analysis
Assessment and validation of concepts and rules
Bibliographical acquisition of knowledge
Formalisation of rules as algorithms
Designer
Objectives
Technical characteristics and objectives
Analysis of existing and/or reference situation
Analysis of the process and its control system (Functional and structural analysis)
Tasks to be performed
Model of Human Activity
Static Tasks allocation Man-Machine
Automatic tasks
Interactive Tasks
Users requirements
Human operator model(limits, resources,..)
A priori modelization of human tasks
Need in assistance
Analysis and choice of decision support tools
Kinds of assistance
Choice the cooperation mode - user /decision support tools
Informational demands
Ergonomic criteria
Human-computer interface specification
Global Human-Machine System generation and integration
Human-Computer Interface
Model of human prescribed tasks
Model of human performed tasks
+
-
Confrontation of both models
variations
Validation/Refinement of Human-computer interace
Cognitive Activity analysis / Data processing
Correlation of the data (reconstitution of the sequence activity)
Analysis of the activity; processing of objective and subjective data
Data collect
Definition of experimental protocols
Real/Simulated Man-Machine System
General of the activity, strategies, limits, resources,
General model of the human operator


Description of « Block Automatique Lumineux » signals functioning

Production of signals

Rules to respect by the experts

Formula used by the experts according to the following items

Comment of the expert

BAL
canton
canton 1500 mètres
c1
c2
c3
8h10
8h15
Tracé de sillon. Toujours ≥ VL anaptin
Equipement en plus normal de circulation.
Tsp = LC1 + LC2 + LTrain / Vitesse x 0,9 + Rhi
, Vitesse
type de convoi.
+ profil
10

**Fig. 9** An expert's written report during a description with the analysts

The *simulation module* is intended to study the feasibility of a base scenario or a scenario modified by a calculation of capacity on a section. To do this, it simulates the movement of the trains, taking into account their dynamic properties, the properties of the track (slopes, ramps, curves....) and the signals.

The *presentation* module makes it possible to visualise the scenarios, capacities and results of the simulation and/or calculation in graphic or alphanumeric form.

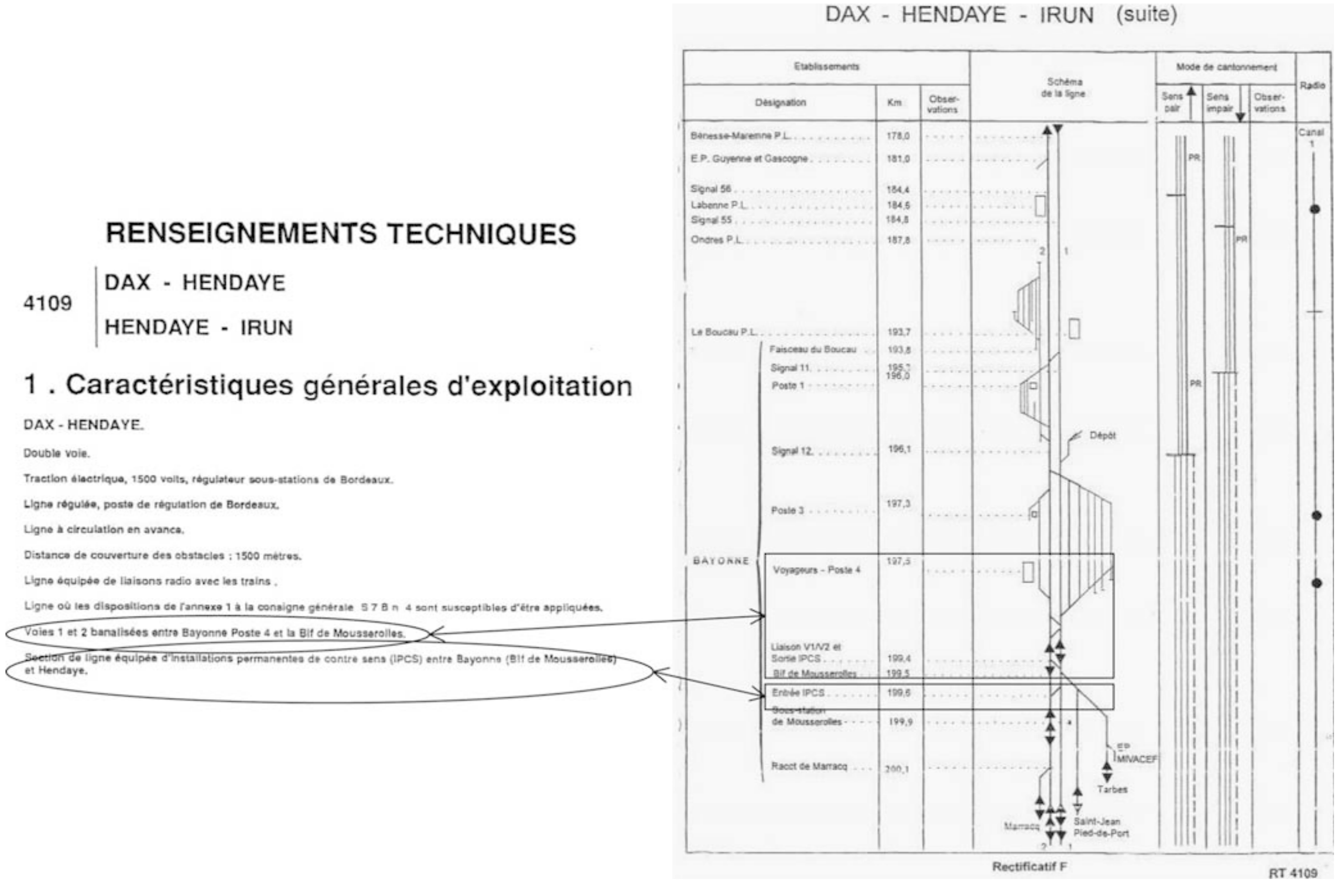
The calculation and simulation modules are independent: the system therefore provides the user with the possibility of launching just the calculation module if he or she has no infrastructure (to be described), or just the simulation module if he or she wishes to check the schedules, or the two can be linked to provide a network capacity obtained by calculation and checked by simulation.

### 4.3.2 Human-machine interfaces

The human-machine interfaces were progressively designed and tested in collaboration with the users. For a greater ease of use, the interfaces are all designed according to the same format. In this way, the user will need less training time and will get used to the interfaces more quickly (Nielsen 1993). For the same reason, the same tool bar will appear in all the interfaces, featuring the classic editing functions such as new, copy, cut, paste and delete, and their symbols.

The first module, shown in Fig. 14, is a "directing" module. Its interface presents the existing studies or allows the creation of new ones; it also enables the user to access the other modules or editors. The interfaces of the editors, such as the reference mark, train, section and scenario editors are very similar. They allow direct access to the respective data so that they can be rapidly consulted or modified. A graphic representation helps the user to become aware of the state of this data.

The calculation module interface offers the user the choice between the various modes of calculation which have been designed for the users (cf. Fig. 15). The user has the choice between several modes: "Auto", "Section", "Journey", "Global" and "Demand". Several of these modes require different parameters to be supplied by the user. The "Auto" mode requires no parameters; the users can use it when they want to have a general idea of the number of additional convoys. The "Section" mode allows the user to enter the journey times for each of the sections, the interest being that these journey times, which



## RENSEIGNEMENTS TECHNIQUES

4109 | DAX - HENDAYE
HENDAYE - IRUN

### 1 . Caractéristiques générales d'exploitation

DAX - HENDAYE.

Double voie.

Traction électrique, 1500 volts, régulateur sous-stations de Bordeaux.

Ligne régulée, poste de régulation de Bordeaux.

Ligne à circulation en avance.

Distance de couverture des obstacles ; 1500 mètres.

Ligne équipée de liaisons radio avec les trains .

Ligne où les dispositions de l'annexe 1 à la consigne générale S 7 B n 4 sont susceptibles d'être appliquées.

Voies 1 et 2 banalisées entre Bayonne Poste 4 et la Bif de Mousserolles.

Section de ligne équipée d'installations permanentes de contre sens (IPCS) entre Bayonne (Bif de Mousserolles) et Hendaye.

DAX - HENDAYE - IRUN (suite)

| Etablissements | | | |
|---|---|---|---|
| | Désignation | Km | Observations |
| | Bénesse-Maremne P.L. | 178,0 | |
| | E.P. Guyenne et Gascogne | 181,0 | |
| | Signal 56 | 184,4 | |
| | Labenne P.L. | 184,5 | |
| | Signal 55 | 184,8 | |
| | Ondres P.L. | 187,8 | |
| | Le Boucau P.L. | 193,7 | |
| | Faisceau du Boucau | 193,8 | |
| | Signal 11 | 195,7 | |
| | Poste 1 | 196,0 | |
| | Signal 12 | 196,1 | |
| | Poste 3 | 197,3 | |
| BAYONNE | Voyageurs - Poste 4 | 197,5 | |
| | Liaison V1/V2 et Sortie IPCS | 199,4 | |
| | Bif de Mousserolles | 199,5 | |
| | Entrée IPCS | 199,6 | |
| | Sous-station de Mousserolles | 199,9 | |
| | Racct de Marracq | 200,1 | |

Rectificatif F

RT 4109

Fig. 10 Documents frequently used by the expert: technical information concerning a railway line

are not representative of existing traffic, could become representative in the future. The ''Journey'' mode enables the user to trace the reference convoy using an existing convoy. The ''Global'' mode distributes the overall journey time over each of the sections. In the ''Demand'' mode, the user can insert several types of additional convoys; these convoys can have different overall journey times and places of departure and/or arrival.

The interface for the simulation module is made up of an action window and a message window (Fig. 16). The action window represents the infrastructure in which the simulation is taking place. The trains are represented by arrows of different colours according to the type of train, which move along the infrastructure. The message window detects all the events which have occurred during the simulation; the following two types of messages can appear:

- Informational messages telling the user how the operations are progressing.
- Warning messages indicating, amongst other things, the absence of departure signals and the crossing of warning signal points.

The error messages show the reasons for the unplanned interruption of the simulation, for example, because of a collision between two trains.

The presentation module interface is made up of a window representing the journeys in the form of a space-time diagram; it includes the opening of another window containing the detailed schedule of a journey (Fig. 17).

## 4.4 Evaluation

The evaluation is intended to check that the interactive decision support system provided for the users meets a set of evaluation criteria established during the first stages. There are several criteria to check: (1) performance criteria; the case study results must be validated in order to show the accuracy and reliability of the results (2) the human-machine interface has to be checked and approved by the users from an ergonomic point of view. The evaluation takes place throughout the process, both upstream and downstream; it corresponds above all to a participatory approach (Muller et al. 1997).

### *4.4.1 An evaluation according to performance criteria*

Two approaches can be used to test the calculation module results. The first uses the experts' opinion. The second involves simulation. The validation is then done by comparing the various results.

We studied the Bordeaux-Hendaye line which was chosen by experts because it is representative of the French railway network. The line is divided into 7

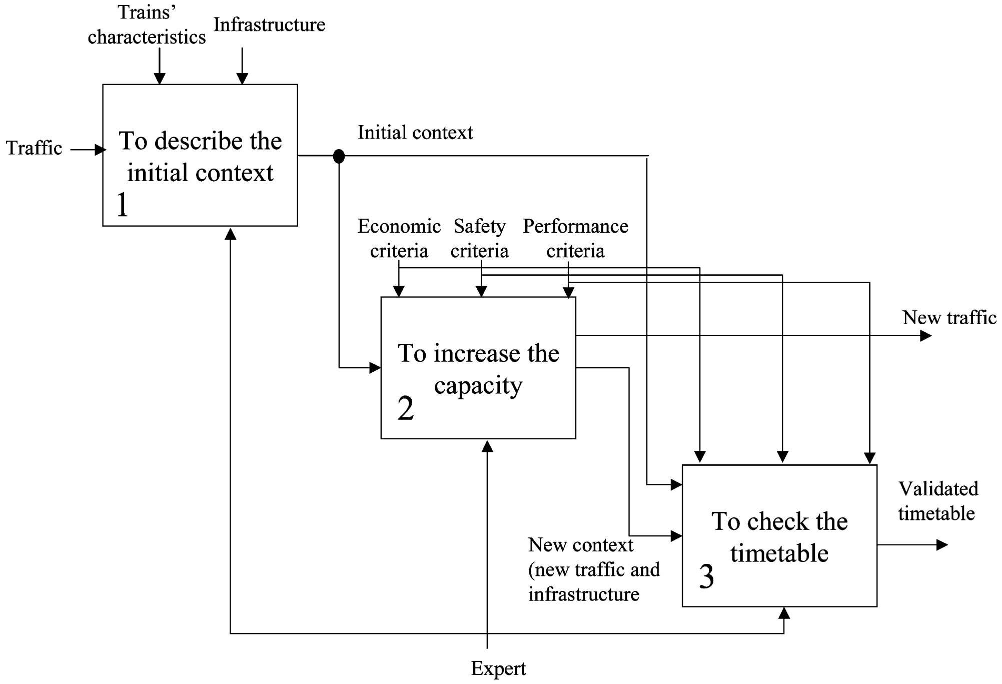


**Fig. 11** Global modelling of the activities performed by the experts

sections. It includes branches via which trains can be inserted or removed from the line; it also has varied spacing between the section limits on departure and arrival. The traffic established is represented by the presentation module on a space-time diagram. We carried out our research for each mode: AUTO, JOURNEY, SECTION, GLOBAL and DEMAND over different periods. For example, after launching the calculation module with the following parameters: section mode, study start at 15:00, study end at 18:00, no stopping time allowed at the section limits, no compulsory stopping time at the section limits, no overtaking allowed, no penalising sections, the system indicates 12 additional convoys. The results are stored in a file which can be used by the presentation module (cf. Fig. 17). The additional convoys and the established convoys can be distinguished in the presentation by different colours; each colour represents a different category of train.

In the INFRAFER system, a simulation module is provided and takes account of all the details concerning the infrastructure and the trains. The simulation is launched with the results from the calculation module in order to check the reliability of the results. This stage is performed in the presence of experts in order to allow them to analyse any errors or the correct functioning of the simulation.

### *4.4.2 An ergonomic evaluation*

The interface was described at the beginning of the project; it then evolved throughout the project in order to meet the user requirements. At the end of the project, the interface must be evaluated (*a posteriori* evaluation) so that it will be completely accepted by the users. Any evaluation consists in identifying or foreseeing difficulties met by the users, in detecting the strong and weak points of the system, and in understanding the reasons for them. Figure 18 shows a sample of the assessment forms filled in with a user during the assessment process. It is essential for the assessors to distinguish between the two evaluation properties for an interface, i.e., usefulness and usability.

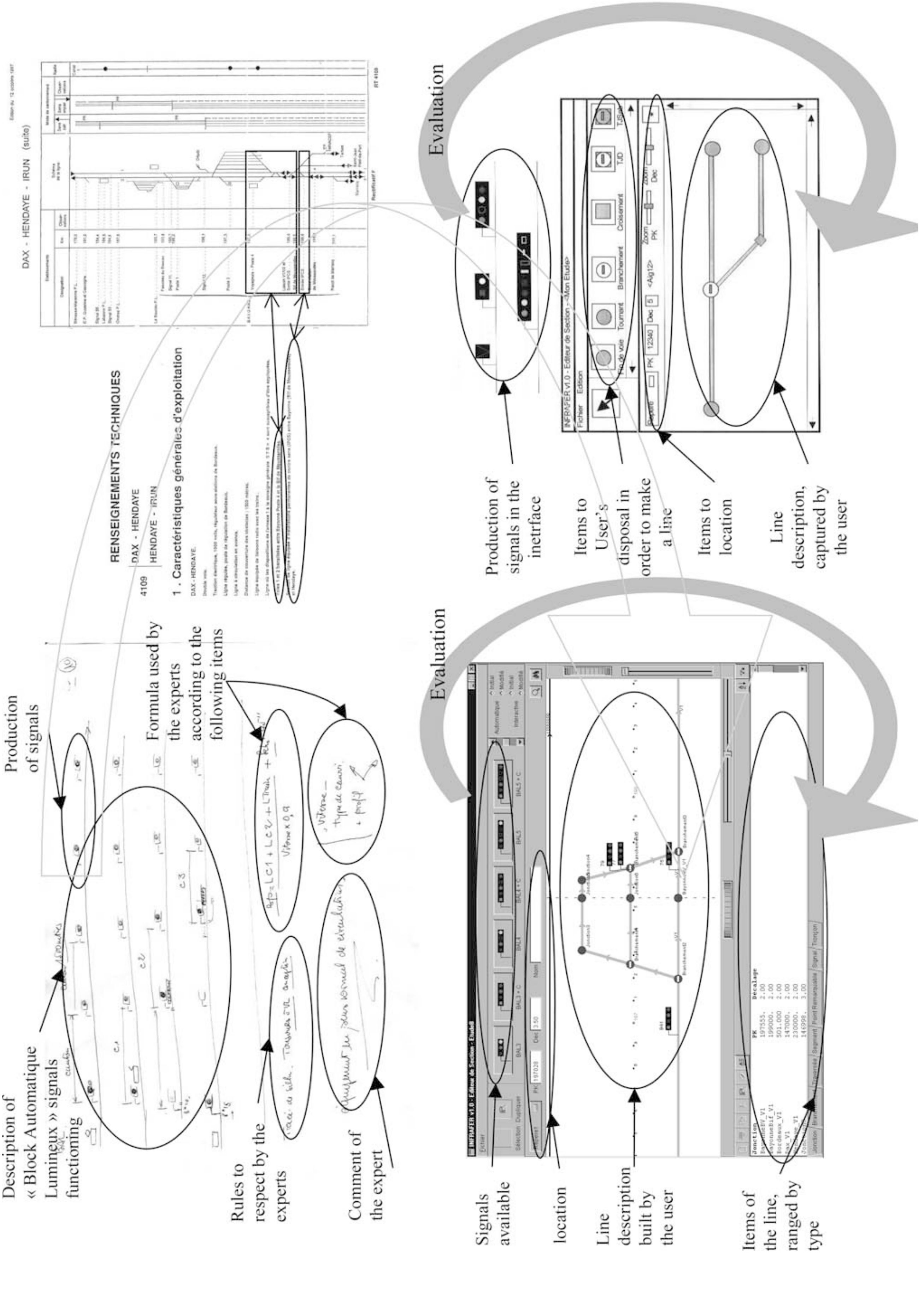
Usefulness can be evaluated by an analysis of a task and/or activities based on two main criteria: task appropriateness and work distribution. Task appropriateness consists in checking whether the cognitive procedures developed by the user are similar to those originally developed by the designer, and thus in estimating whether the task which has been redefined by the user is compatible with the task to be performed. The distribution of tasks was decided in the descending phase of the U-model. It is therefore a question of checking whether the model obtained really corresponds to the expected model during the various studies performed by experts when using the system.

As regards the ergonomic evaluation, there is a series of well-known criteria and heuristics which aim at helping the assessor to estimate the ergonomic quality of the interface and to take decisions if necessary concerning modifications and/or improvements. Amongst all the existing evaluation methods, we selected an empirical approach which appeared to be the best adapted to our work. This method is widespread in the evaluation of user interfaces (Nielsen et al. 1994, 1990) .

Because of lack of space and also because the evaluations are performed in parallel to the design process, it

►

**Fig. 12** The progress of designing the interface


Description of « Block Automatique Lumineux » signals functioning
Production of signals
Formula used by the experts according to the following items
Rules to respect by the experts
Comment of the expert
Signals available
location
Line description built by the user
Items of the line, ranged by type
Evaluation
Evaluation
Production of signals in the inetrface
Items to User's disposal in order to make a line
Items to location
Line description, captured by the user
RENSEIGNEMENTS TECHNIQUES
4109
DAX - HENDAYE
HENDAYE - IRUN
1. Caractéristiques générales d'exploitation
DAX - HENDAYE - IRUN (suite)
Vitesse x 0,9
INFRAFER v1.0 - Editeur de Section - Etude
INFRAFER v1.0 - Editeur de Section - <Mon Etude>
Fichier
Edition
BAL3
BAL3 + C
BAL4
BAL4 + C
BAL5
BAL5 + C
Automatique
Interactive
Fin de voie
Tournant
Branchement
Croisement
TJD
TJS
PK
Dec
Zoom
Jonction
Traversée
Segment
Point Remarquable
Signal
Tronçon

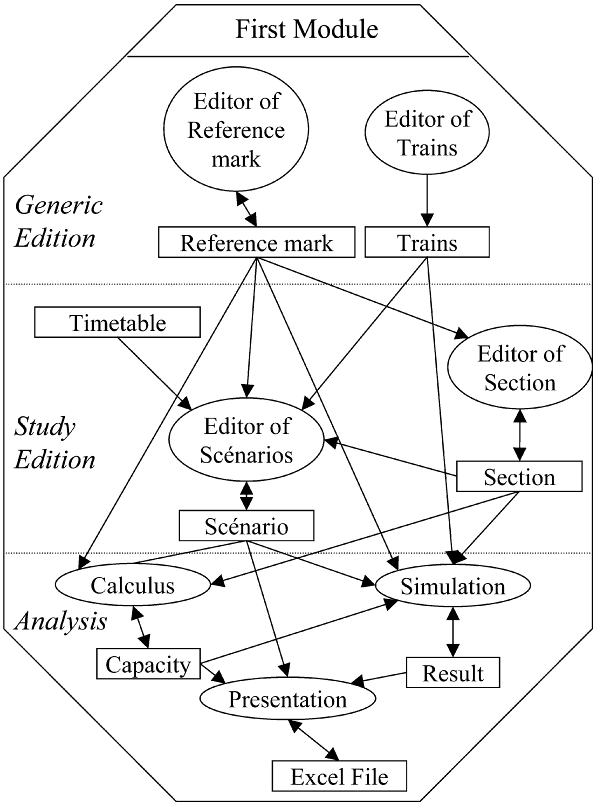
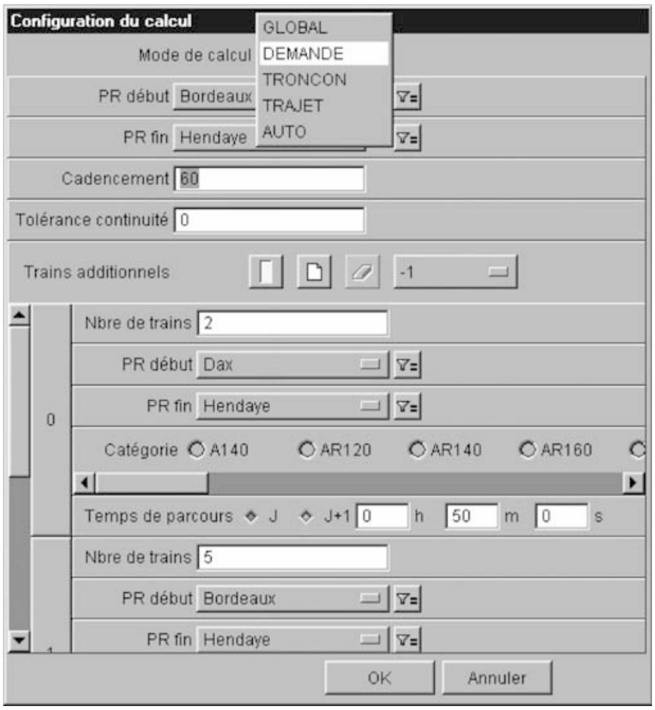


**Fig. 13** A diagram showing the links between INFRAFER modules

is impossible to go into the details of the evaluations here. The most important point is no doubt the insistence of the experts (users of the system) throughout the project concerning the question of avoiding to a maximum the problems of human-machine interaction and repetitive infrastructure description tasks in order to concentrate on their railway investment research projects. The fine adjustments of the human-machine interfaces therefore went in this direction.


Configuration du calcul

Mode de calcul: GLOBAL / DEMANDE / TRONCON / TRAJET / AUTO

PR début: Bordeaux

PR fin: Hendaye

Cadencement: 60

Tolérance continuité: 0

Trains additionnels: -1

0
- Nbre de trains: 2
- PR début: Dax
- PR fin: Hendaye
- Catégorie: ○ A140 ○ AR120 ○ AR140 ○ AR160
- Temps de parcours: ◆ J ◇ J+1 0 h 50 m 0 s

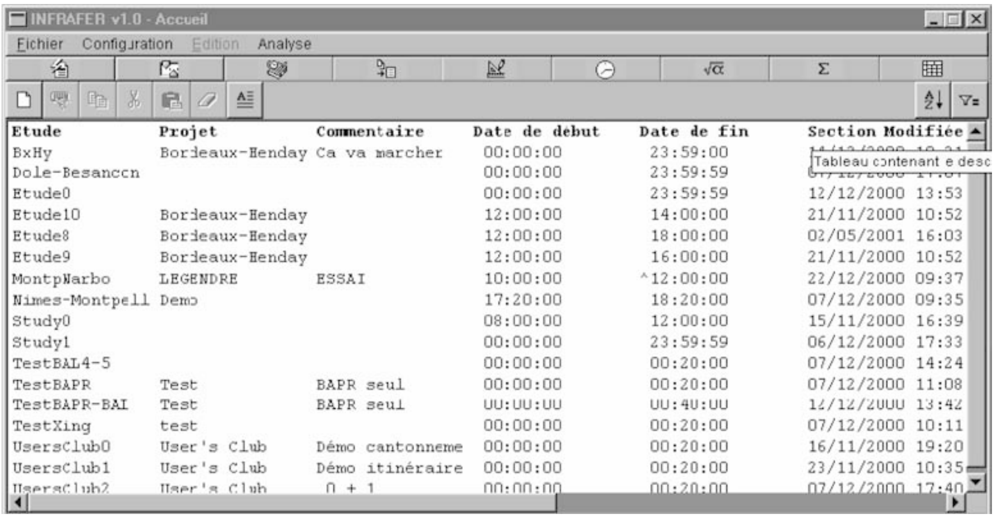
1
- Nbre de trains: 5
- PR début: Bordeaux
- PR fin: Hendaye

**Fig. 15** The journey configuration window

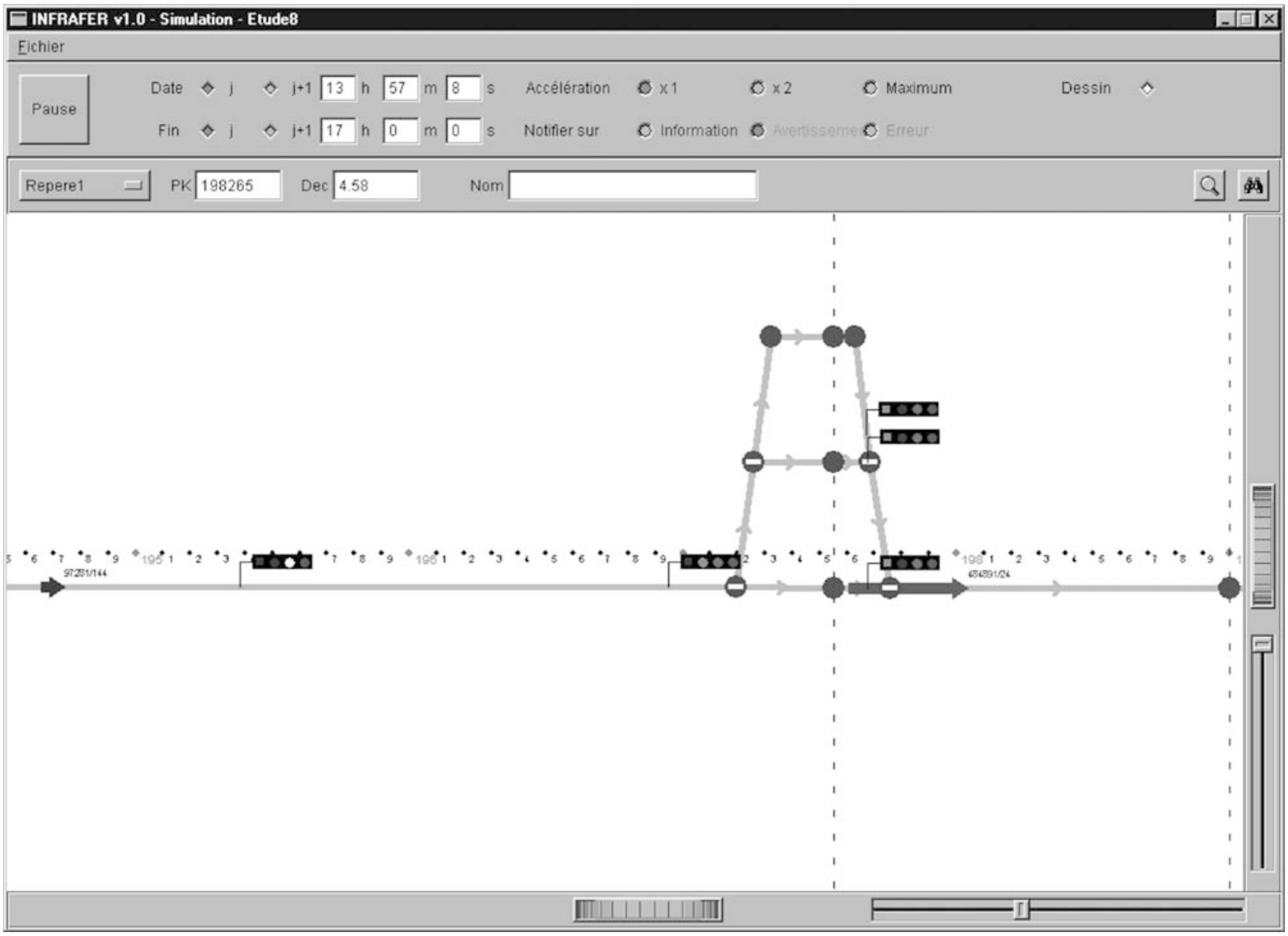
## 5 Conclusions

One great difficulty in human-machine systems concerns the methodological process to be applied to design and evaluate a system. The U-model has been described in detail in this article. Above all, the U-model provides a multidisciplinary study framework for the participants in a project of the design and evaluation of an interactive system in an industrial context, implying a certain complexity. This global model is intended to be generic. Thus, from one industrial application to another,

**Fig. 14** The first window of INFRAFER



| Etude | Projet | Commentaire | Date de debut | Date de fin | Section Modifiée |
|---|---|---|---|---|---|
| BxHy | Bordeaux-Henday | Ca va marcher | 00:00:00 | 23:59:00 | [illegible] |
| Dole-Besancon | | | 00:00:00 | 23:59:59 | [illegible] |
| Etude0 | | | 00:00:00 | 23:59:59 | 12/12/2000 13:53 |
| Etude10 | Bordeaux-Henday | | 12:00:00 | 14:00:00 | 21/11/2000 10:52 |
| Etude8 | Bordeaux-Henday | | 12:00:00 | 18:00:00 | 02/05/2001 16:03 |
| Etude9 | Bordeaux-Henday | | 12:00:00 | 16:00:00 | 21/11/2000 10:52 |
| MontpNarbo | LEGENDRE | ESSAI | 10:00:00 | ^12:00:00 | 22/12/2000 09:37 |
| Nimes-Montpell | Demo | | 17:20:00 | 18:20:00 | 07/12/2000 09:35 |
| Study0 | | | 08:00:00 | 12:00:00 | 15/11/2000 16:39 |
| Study1 | | | 00:00:00 | 23:59:59 | 06/12/2000 17:33 |
| TestBAL4-5 | | | 00:00:00 | 00:20:00 | 07/12/2000 14:24 |
| TestBAPR | Test | BAPR seul | 00:00:00 | 00:20:00 | 07/12/2000 11:08 |
| TestBAPR-BAI | Test | BAPR seul | 00:00:00 | 00:40:00 | 12/12/2000 13:42 |
| TestXing | test | | 00:00:00 | 00:20:00 | 07/12/2000 10:11 |
| UsersClub0 | User's Club | Démo cantonneme | 00:00:00 | 00:20:00 | 16/11/2000 19:20 |
| UsersClub1 | User's Club | Démo itinéraire | 00:00:00 | 00:20:00 | 23/11/2000 10:35 |
| UsersClub2 | User's Club | 0 + 1 | 00:00:00 | 00:20:00 | 07/12/2000 17:40 |



INFRAFER v1.0 - Simulation - Etude8

Fichier

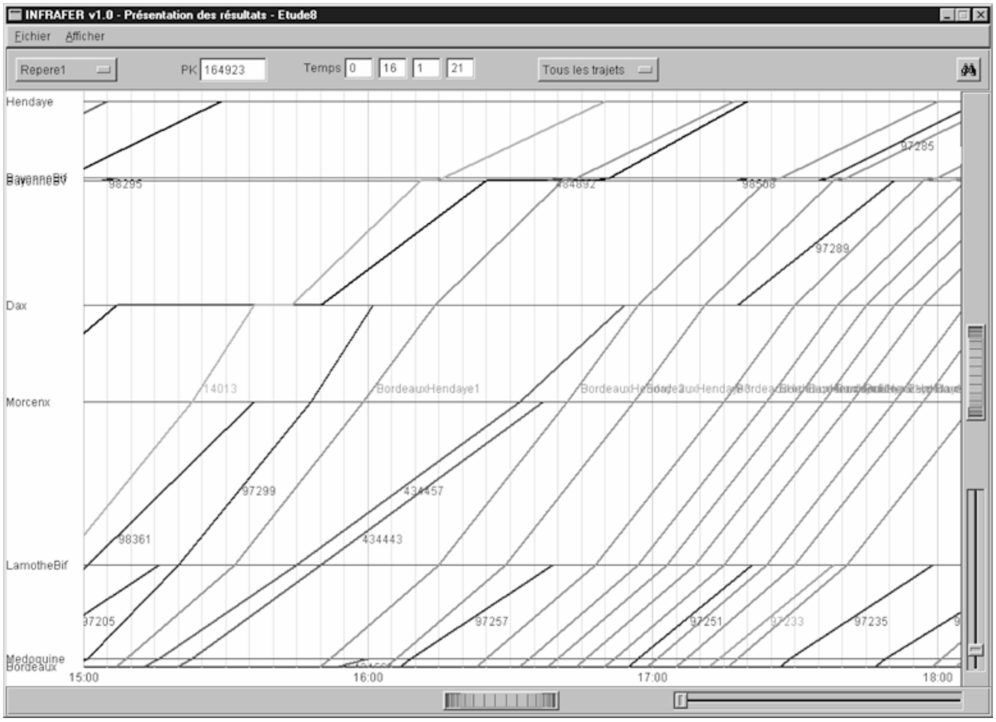
Pause

Date ◆ j ◇ j+1 13 h 57 m 8 s Accélération ○ x 1 ○ x 2 ○ Maximum Dessin ◇

Fin ◆ j ◇ j+1 17 h 0 m 0 s Notifier sur ○ Information ○ Avertissement ○ Erreur

Repere1 PK 198265 Dec 4.58 Nom

Fig. 16 The simulation window

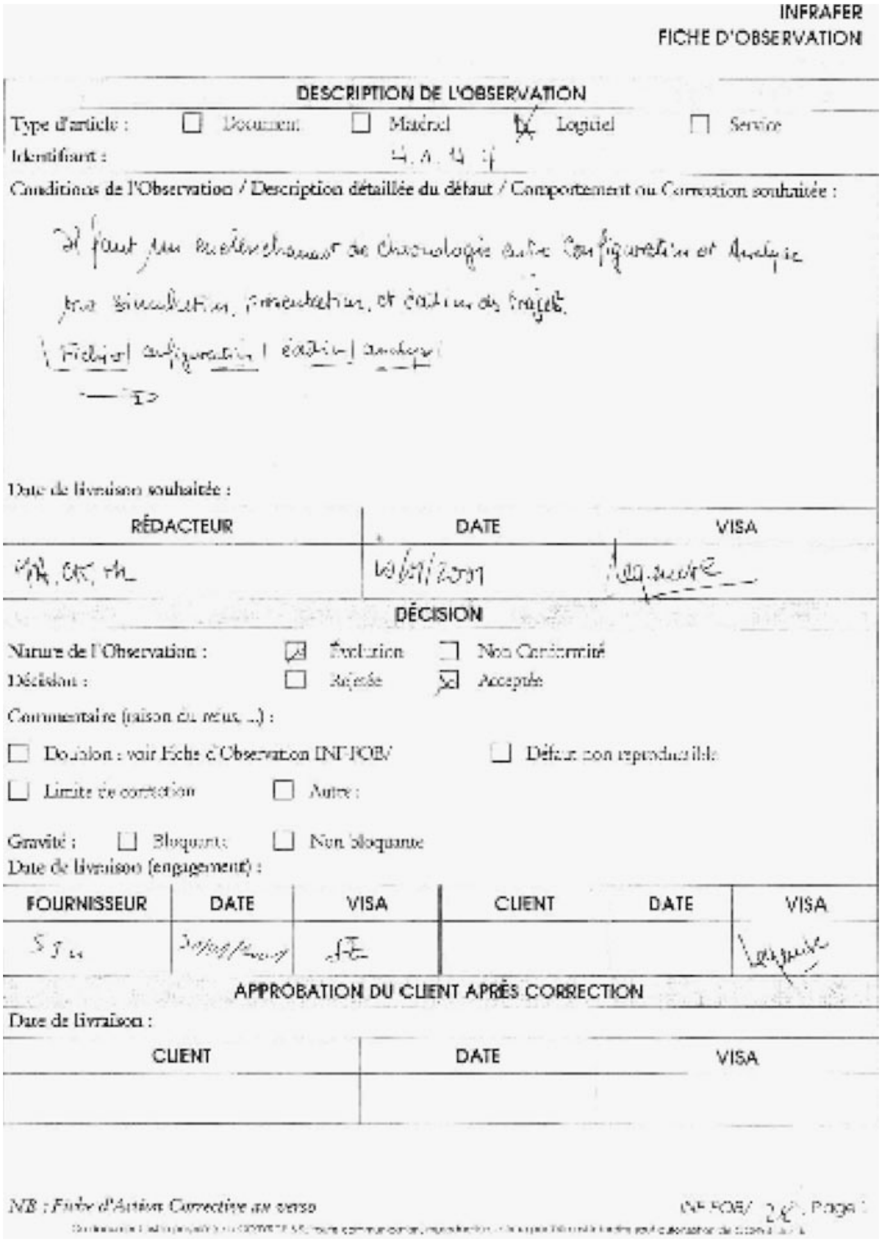


INFRAFER v1.0 - Présentation des résultats - Etude8

Fichier Afficher

Repere1 PK 164923 Temps 0 16 1 21 Tous les trajets

Fig. 17 A presentation of the results



INFRAFER
FICHE D'OBSERVATION

DESCRIPTION DE L'OBSERVATION

Type d'article : □ Document □ Matériel ☒ Logiciel □ Service

Identifiant : 4.1.4.1

Conditions de l'Observation / Description détaillée du défaut / Comportement ou Correction souhaitée :

Il faut un enchaînement de chronologie entre Configuration et Analyse pour simulation, présentation, et édition des trajets.

\ Fichier | configuration | édition | analyse \

—▷

Date de livraison souhaitée :

| RÉDACTEUR | DATE | VISA |
|---|---|---|
| MR, CK, HL | 10/07/2001 | Legendre |

DÉCISION

Nature de l'Observation : ☒ Évolution □ Non Conformité

Décision : □ Rejetée ☒ Acceptée

Commentaire (raison du refus, ...) :

□ Doublon : voir Fiche d'Observation INF-FOBV □ Défaut non reproductible

□ Limite de correction □ Autre :

Gravité : □ Bloquante □ Non bloquante

Date de livraison (engagement) :

| FOURNISSEUR | DATE | VISA | CLIENT | DATE | VISA |
|---|---|---|---|---|---|
| SJu | 20/09/2001 | SJ | | | Legendre |

APPROBATION DU CLIENT APRÈS CORRECTION

Date de livraison :

| CLIENT | DATE | VISA |
|---|---|---|
| | | |

*NB : Fiche d'Action Corrective au verso* INF-FOBV 26 Page 1

[illegible]

**Fig. 18** The assessment form filled in by a user during the assessment process

according to its specificities, it can be necessary to adapt the process effectively.

In this article, we have explained how it has been used in the context of the design and evaluation of an interactive decision support system in a railway investment (which can cost millions of euros). One of the centres of interest of this project is that the users of the system targeted are experts in their field, who require support which closely follows their strategies and helps them correctly in their decision-making processes; this support meets needs in simulation as well as in calculation. The U-model was markedly suitable in relation to this specificity.

**Acknowledgements** The authors would like to thank the anonymous reviewers of this paper for their constructive comments, as well as Michel Legendre (RFF, Paris), Serge Jung, Gérard Paulhac and Jacques Moncorgé (CORYS TESS, Grenoble) for the fruitful exchanges throughout the INFRAFER project, leading to the design and evaluation of the system. The authors would also like to thank Caroline Guillaume-Blaydes for her help in translating this paper.

## References

Abed M (1990) Contribution à la modélisation de la tâche par outils de spécification exploitant les mouvements oculaires: application à la conception et à l'évaluation des interfaces homme-machine. Dissertation, Université de Valenciennes et du Hainaut-Cambrésis, France

Abed M (2001) Méthodes et modèles formels et semi-formels pour la conception et l'évaluation des systèmes homme-machine. Habilitation thesis, Université de Valenciennes et du Hainaut-Cambrésis, France

Abed M, Angué JC (1990) Using the measure of eye movements to modelize an operator's activity. In: Proceedings of the Ninth European Annual Conference on Human Decision Making and Manual Control, Varese, Italy, 10–12 September 1990

Abed M, Angué JC (1994) A new method for conception, realisation and evaluation of man-machine. In: Proceedings of the International Conference on System, Man and Cybernetics, IEEE SMC-94, San Antonio, TX, 2–5 October 1994

Abed M, Bernard JM, Angue JC (1991) Task analysis and modelization by using SADT and PETRI networks. In: Proceedings of the Tenth European Annual Conference on Human Decision Making and Manual Control, Liège, Belgium, 11–13 November 1991

Amalberti R (1997) Human error in aviation. In: Proceedings of the International Aviation Safety Conference (IASC-97), Utrecht, 27–29 August 1997

Arlat J (1995) Guide de la sûreté de fonctionnement. Cépaduès, Toulouse

Bainbridge L (1978) The process controller. In: Singleton WT (ed) The analysis of practical skills, M.T.P. Press, Lancaster, UK

Balbo S (1994) Evaluation ergonomique des interfaces utilisateur: un pas vers l'automatisation. Dissertation, Université de Grenoble

Bastien JM, Scapin DL (2001) Evaluation des systèmes d'information et critères ergonomiques. In: Kolski C (ed) Environnements évolués et évaluation de l'IHM, Interaction Homme Machine pour les SI, Vol. 2. Éditions Hermes, Paris

Bernard JM (1994) Exploitation des mesures oculométriques dans la modélisation de la tâche prescrite et de l'activité réelle des opérateurs par réseaux de Petri. Dissertation, Université de Valenciennes et du Hainaut-Cambrésis

Bodart F, Hennebert AM, Leheureux JM, Provot I, Sacré B and Vanderdonckt J (1995) Towards a systematic building of software architectures: the trident methodological guide. In: Proceedings of the Eurographics Workshop on Design, Specification, and Verification of Interactive Systems (DSV-IS'95), Château de Bonas, France, 7–9 June 1995

Boehm BW (1981) Software engineering economics. Prentice-Hall, Englewood Cliffs, NJ

Boehm BW, Gray TE and Seewaldt T (1984) Prototyping versus specifying: a multiproject experiment. IEEE Trans Soft Engin 10(3):290–303

Booch G (1994) Conception orientée objets et applications, 2nd edn. Addition-Wesley, Reading, MA

Brun P (1998) XTL: une logique temporelle pour la spécification formelle des systèmes interactifs. Dissertation, Université Paris XI Orsay

Buisine A (1999) Vers une démarche industrielle pour le développement d'interfaces homme-machine. Dissertation, Université de Rouen, France

Cacciabue PC, Decortis F, Drozdowicz B, Masson M and Nordvik JP (1992) COSIMO: a cognitive simulation model of human decision making and behavior in accident management of complex plants. IEEE Trans Sys Man Cybern 22(5):1058–1074

Coutaz J (1995) Interaction homme-machine: points d'ancrage entre ergonomie et génie logiciel. In: Proceedings of Design'95, 4ème table ronde francophone sur la conception, Autrans, France, 11–13 January 1995

Curtis B, Hefley B (1994) A WIMP no more, the maturing of user interface engineering. Interactions 1(1):22–34

David R, Alla H (1994) Petri nets for modeling dynamic systems, a survey. Automatica 30:175–202

De Keyser V, Decortis F, Housiaux A and Van Daele A (1987) Les communications homme-machine dans les systèmes complexes. Rapport politique scientifique FAST no. 8, University of Liège, Belgium

DeMarco T (1979) Structured analysis and system specification. Yourdon Press, Englewood Cliffs, NJ—Prentice–Hall, New York

Dix A, Finlay J, Abowd G and Beale R (1998) Human-computer interaction, second edition. Prentice-Hall, Englewood Cliffs, NJ

ESA (1991) Software engineering standards, ESA PSS-05–0. Issue 2, February, European Space Agency

Ezzedine H, Abed M (1997) Une méthode d'évaluation d'interface homme-machine de supervision d'un procédé industriel. JESA 31(7):1078–1110

Fadier E (1990) Fiabilité humaine: méthodes d'analyse et domaines d'application. In: Leplat J, De Terssac G (eds) Les facteurs humains de la fiabilité dans les systèmes complexes, Edition Octarés, Marseille

Farenc C, Barthet MF and Liberati V (1996) Automatic ergonomic evaluation: which are the limits. In: Proceedings of the 2nd International Workshop on Computer-Aided Design of User Interfaces (CADUI'96), Namur, Belgium, 5–7 June 1996

Fekete JD, Girard P (2001) Environnements de développement de systèmes interactifs. In: Kolski C (ed) Environnements évolués et évaluation de l'IHM, Interaction homme-machine pour les SI 2, Hermes, Paris, pp 23–52

Florio L, Mussone L (1995) A method of capacity computation for complex railway systems. In: Proceedings of the World Conference on Transport Research, Sydney, Australia, 16–21 July 1995

Florio L, Mussone L (1998) An analytical model for the simultaneous calculation of capacity of lines, junctions and station tracks. In: Mellit B, Hill RJ, Allan J, Scuittio G and Brebbia CA (eds) Computers in railways VI, WIT Press, Wessex, UK

Fontaine M, Gauyacq D (2001) SISYFE: a toolbox to simulate the railway network functioning for many purposes. Some cases of application. In: Proceedings of the World Congress on Railway Research, Cologne, Germany, November 2001

Fréjus M (1999) Analyser l'activité d'explication pour concevoir en terme d'aide: application à la formation et à la négociation commerciale. Dissertation, Université Paris 5

Gamboa-Rodrìguez F, Scapin D (1997) Editing MAD* task descriptions for specifying interfaces at both semantic and presentation levels. In: Proceedings of DSV-IS'97, Springer, Berlin Heidelberg New York

Grislin M, Kolski C (1997) Proposition d'une démarche d'évaluation a priori des interfaces homme-machine: application aux étapes de spécification fonctionnelle et de conception préliminaire. RAIRO-APII-JESA 31:1111–1154

Grudin J (1992) Utility and usability: research issues and development contexts. Interact Comp 4(2):209–217

Hachemane P (1997) Evaluation de la capacité de réseaux ferroviaires. Dissertation, Ecole polytechnique Fédérale de Lausanne

Harrison MD, Duke DJ (1994) A review of formalisms for describing interactive behaviour. In: Proceedings of ICSE'94, the Workshop on Software Engineering and Human Computer Interaction, Sorrento, Italy, 16–17 May 1994

Hartson HR, Boehm-Davis D (1993) User interface development processes and methodologies. Behav Info Technol 12(2):98–114

Hassl DF (1965) Advanced concepts in fault tree analysis. In: Proceedings of the System Safety Symposium, Seattle, WA, 8–9 June 1965

Hatley DJ, Pirbhai IA (1991) Stratégies de spécification des systèmes temps réel (SA-RT). Editions Masson, Paris

Helander M, Landauer TK and Prabhu P (eds) Handbook of human-computer interaction. Elsevier, New York

Hix D (1995) Usability evaluation: how does it relate to software engineering? In: Anzai Y, Ogawa K and Mori H (eds) Symbiosis of human and artifact: human and social aspects of human-computer interaction, Elsevier, New York

Hix D, Hartson HR (1993) Developing user interface: ensuring usability through product & process. Wiley, New York

Hoc JM, Samurcay R (1992) An ergonomic approach to knowledge representation. Reliab Engin Sys Safe 36:217–230

Hollnagel E, Cacciabue PC (1999) Cognition, technology & work: an introduction. Cog Technol Work 1:1–6

Jambon F, Brun P, Aït-Ameur Y (2001) Spécification des systèmes interactifs. In: Kolski C (ed) Analyse et conception de l'IHM, Interaction Homme Machine pour les SI, Vol. 1, Paris: Éditions Hermes

Jaulent P (1990) Génie logiciel, les méthodes. Armand Colin, Paris

Johnson H, Johnson P (1991) Task knowledge structures: psychological basis and integration into system design. Acta Psychologica 78:3–26

Johnson P (1999) Theory based design: from individual users and tasks to collaborative systems. In: Vanderdonckt J, Puerta A (eds) Computer-aided design of user interfaces II (CADUI'99), Kluwer, Amsterdam

Kolski C (1997) Interfaces homme-machine application aux systèmes industriels complexes. Hermes, Paris

Kolski C (1998) A call for answers around the proposition of an HCI-enriched model. ACM SIGSOFT SE Notes 3:93–96

Kolski C, Millot P (1991) A rule-based approach to the ergonomic "static" evaluation of man-machine graphic interface in industrial processes. Int J Man-Mach Stud 35:657–674

Kolski C, Riera B and Berger T (2000). A questionnaire-based discount evaluation method using guidelines for process control interactive applications. In: Vanderdonckt J, Farenc C (eds) Proceedings of Tools for Working With Guidelines TFWWG'2000, Springer, Berlin Heidelberg New York

Kolski C, Ezzedine H and Abed M (2001) Développement du logiciel: des cycles classiques aux cycles enrichis sous l'angle des IHM. In: Kolski C (ed) Analyse et conception de l'IHM, Interaction Homme-machine pour les SI, Hermes, Paris

Lepreux S, Abed M, Kolski C, Jung S and Legendre M (2001) A methodology for decision support system design in railway capacity evaluation. In: Morten L (ed) Proceedings of the 20th European Annual Conference on Human Decision Making and Manual Control, EAM2001, Copenhagen, Denmark, 25–27 June 2001

Lichter H, Schneider-Hufschmidt M and Zullighoven H (1994) Prototyping in industrial software projects: bridging the gap between theory and practice. IEEE Trans Sort Engin 20(11):825–832

Lim KY, Long JB (1994) The MUSE method for usability engineering. Cambridge series on human-computer interaction, Cambridge University Press, Cambridge, UK

Lind M (1990) Representing goals and functions of complex systems: an introduction to multilevel flow modelling. In: Proceedings of the 2nd European Meeting on Cognitive Science Approaches to Process Control, Siena, Italy, October 1990

Lu S, Paris C and Van der Linden K (1999) Towards the automatic generation of task models from object oriented diagrams. In: Dewan P, Chatty S (eds) Proceedings of the IFIP Working Conference on Engineering for Human-Computer Interaction, Heraklion, Crete, Greece, 14–18 September 1999, Kluwer, pp 169–190

Macaulay LA (1996) Requirements engineering. Springer, Berlin Heidelberg New York

Maguire M (2001) Methods to support human-centred design. Int J Hum Comp Stud 55:587–634

Marca DA, McGowan CL (1988) SADT: structured analysis and design technique. McGraw-Hill, New York

McDermid J, Ripkin K (1984) Life cycle support in the ADA environment. Cambridge University Press, Cambridge, UK

McGee S, Howard B (1998) Evaluating educational multimedia in the context of use. J Univ Comp Sci 4:273–291

McKenna S (1996) Evaluating IMM: issues for researchers. Open Learning Institute, 1996. http://www.csu.edu.au/division/OLI. Cited 1996

Millot P (1988) Supervision des procédés automatisés et ergonomie. Editions Hermes, Paris

Millot P, Debernard S (1993) Men-machines cooperative organizations: methodologicial and practical attempts in air traffic control. In: Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Le Touquet, France, 17–20 October 1993

Millot P, Roussillon E (1991) Man-machine cooperation in telerobotics: problematics and methodologies. In: Proceedings of the Second Symposium on Robotics, Institut National des Sciences et Techniques Nucléaires, Gif sur Yvette, France, April 1991

Moray N (1997) Human factors in process control. In: Salvendy G (ed) Handbook of human factors and ergonomics, Wiley, New York

Moussa F, Kolski C and Riahi M (2000) A model based approach to semi-automated user interface generation for process control interactive applications. Interact Comp 12:(3):245–279

Muller MJ, Haslwanter JH and Dayton T (1997) Participatory practices in the software lifecycle. In: Helander M, Landauer TK, Prabhu P (eds) Handbook of human-computer interaction, Elsevier, New York

Myers BA (1993) State of the art in user interface software tools. In: Hartson HR and Hix D (eds) Advances in human-computer interaction, Ablex Publishing, Norwood, NJ

Myers BA (1995) User interface software tools. ACM Trans Comp-Hum Inter 2:64–103

Nardi B (ed) Context and consciousness: activity theory and human-computer interaction. MIT Press, Cambridge, MA

Navarre D, Palanque P and Bastide R (2003) Beyond modelling: a formal and tool supported technique for the iterative and synergistic modelling and cross validation of tasks and system. Interact Comp (in press)

Nielsen J (1993) Usability engineering. Academic, Boston, MA

Nielsen J, Mack RL (1994) Usability inspection methods. Wiley, New York

Nielsen J, Molich R (1990) Heuristic evaluation of user interfaces. In: Proceedings of the CHI'90 Conference on Human-Computer Interaction, Seattle, WA, April 1990

Norman DA (1986) Cognitive engineering. In: Norman DA, Draper SW (eds) User centred system design: new perspectives on human computer interaction, Erlbaum, Hillsdale, NJ

Olson JR, Rueter HH (1987) Extracting expertise from experts: methods for knowledge acquisition. Exp Sys 4(3):152–168

Palanque P (1997) Spécification formelles et systèmes interactifs: vers des systèmes fiables et utilisables. Habilitation, l'Université de Toulouse I

Paternò F (2000) Model-based design and evaluation of interactive applications. Springer, Berlin Heidelberg New York

Paternò F, Mancini C and Meniconi S (1997) ConcurTaskTress: a diagrammatic notation for specifying task models. In: Proceedings of Interact'97, Chapman & Hall, London

Paulhac G, Jung S, Abed M, Kolski C, Lepreux S and Legendre M (2001) Outil d'aide à la conception des infrastructures ferroviaires. Final research report, PREDIT project, CORYS-LA-MIH-RFF

Poltrock S, Grundin J (1995) A survey of systems and behavioural issues. In: Proceedings of the Conference on Human Factors in Computing Systems, Denver, CO, 7–11 May 1995

Preece J, Rogers Y, Sharp H, Benyon D, Holland S and Carey T (1994) Human-computer interaction. Addison-Wesley, Reading, MA

Rasmussen J (1986) Information processing and human-machine interaction, an approach to cognitive engineering. Elsevier, New York

Reason J (1988) Cognitive aids in process environments: prostheses or tools? In: Hollnagel E, Mancini G and Woods DW (eds) Cognitive engineering in complex dynamic worlds, Academic, London

Recht JL (1966) Failure mode and effect. National Safety Council, USA

Rubin KS, Jones PM and Mitchell CM (1988) OFMspert: inference of operator intentions in supervisory control using a blackboard architecture. IEEE Trans Sys Man Cybern 18:618–637

Scapin DL, Bastien JMC (1997) Ergonomic criteria for evaluating the ergonomic quality of interactive systems. Behav Info Technol 16:220–231

Scapin DL, Pierret-Golbreich C (1990) Towards a method for task description: MAD. In: Berlinguet L, Berthelette D (eds) Proceedings of Work with Display Units 89, Elsevier, New York

Sebillotte S (1995) Methodology guide to task analysis with the goal of extracting relevant characteristics for human-computer interfaces. Int J Hum Comp Interact 7:341–363

Senach B (1990) Evaluation ergonomique des interfaces homme-machine: une revue de la literature. Research report, INRIA

Shackel B (1991) Usability, context, framework, definition, design and evaluation. In: Shackel B, Richardson S (eds) Human factors for informatics usability, Cambridge University Press, Cambridge, UK

Singleton WT (1974) Man-machine systems. Penguin, Harmondsworth, UK

Smith SL, Mosier JN (1986) Guidelines for designing user interface software. Report EDS-TR-86–278, The MITRE Corporation, Bedford, MA

SOFRERAIL (1992) SOFTIME computer aided timetable design. Research report, SOFRERAIL

Sommerville I (1994) Le génie logiciel, 4ème édition. Addison-Wesley, Reading, MA

Sperandio JC (1991) Les méthodes d'analyse du travail en psychologie ergonomique. In: Rossi JP (ed) La recherche en psychologie (domaines et méthodes), Bordas, Paris

Szekely P (1996) Retrospective and challenges for model-based interface development. In: Proceedings of the Eurographics Workshop, Namur, Belgium, 5–7 June 1996

Tabary D (2001) Contribution à TOOD, une méthode à base de modèles pour la spécification et la conception des systèmes interactifs. Dissertation, Université de Valenciennes et du Hainaut-Cambrésis

Tarby JC, Barthet MF (1996) The DIANE+ method. In: Vanderdonckt J (ed) Proceedings of the 2nd International Workshop on Computer-Aided Design of User Interfaces (CADUI'96), Presses Universitaires de Namur, Namur, Belgium

Thayer RH, McGettrick AD (eds) Software engineering: a European perspective. IEEE Computer Society Press, New York

Theureau J (1992) Le cours d'action: analyse sémio-logique. Essai d'une anthropologie cognitive située. Peter Lang, Bern, Switzerland

Theureau J, Jeffroy F (1994) Ergonomie des situations informatisées: la conception centrée sur le cours d'action. Editions Octarès, Toulouse, France

Tijus CA, Poiternaud S (1996) Propriétés, objets, procédures: les réseaux sémantiques d'action appliqué à la représentation des dispositifs techniques. Le Travail Humain 59:209–229

UIC (1979) Méthode destinée à determiner la capacité de lignes, code 405, Research Report, UIC

Van-eylen H, Hiraclides G (1996) GRAAL: en quête d'une démarche de développement d'interface utilisateur. Collection Méthode, Angkor Editions, Paris

Vanderdonckt J (1994) Guide ergonomique des interfaces homme-machine. Presses Universitaires de Namur, Namur, Belgium

Villemeur A (1992) Reliability, availability, maintainability and safety assessment (vols.1 and 2). Wiley, Chichester, UK

Wehner T, Clases H and Bachmann C (2000) Cooperation at work: a process-oriented perspective on joint activity in inter-organizational relations. Ergonomics 43:983–998

Wilson JR, Corlett EN (eds) Evaluation of human works: a practical ergonomics methodology, 2nd edn. Taylor & Francis, London

Woods DD (1986) Cognitive technologies: the design of joint human-machine systems. AI 6(4):86–92