RAMSES: A Method for the Design Process of Interactive Information Systems

A. Mouloudi; P. Morizet-Mahoudeaux; A. Valentin

* UMR-CNRS 6599, Heudiasyc, University of Compiègne, France
* JE 2460 ODIC, University of Compiègne, France

Online publication date: 15 January 2011


To link to this Article DOI: 10.1080/10447318.2010.502101

URL: http://dx.doi.org/10.1080/10447318.2010.502101

PLEASE SCROLL DOWN FOR ARTICLE
RAMSES: A Method for the Design Process of Interactive Information Systems

A. Mouloudi¹, P. Morizet-Mahoudeaux¹, and A. Valentin²
1UMR-CNRS 6599, Heudiasyc, University of Compiègne, France
2JE 2460 ODIC, University of Compiègne, France

This work presents a method that aims at structuring the design process of interactive information systems (IIS). This method, which takes parts of its ground in the joint cognitive systems approach, formalizes the integration of usage data in the design process of IIS. Data are generated by using methods of activity analysis. The assumption is that for integrating usage data successfully, they must be represented in a formalism close to the development language, while maintaining the formalization language accessible to each domain specialist involved in the design process. The method can be split up into five steps: (a) data collection of users’ needs in different contexts, (b) representation of the collected data, (c) modeling of corresponding knowledge in an implementation formalism, (d) specification of the IIS, and (e) validation of the model by a functional specification aid tool. A functional specification design tool has been developed according to the presented approach, and has been applied to the design of a traveler information system.

1. INTRODUCTION

A central problem for interactive information systems (IISs) is designing for effective interaction. An IIS is one where the user, as well as being a data resource, is a recipient of the information produced. The purpose of the IIS is to provide the user with information that meets the user’s needs. Integration of users’ needs is a central and organizing process in IIS design (Belkin & Vickery, 1985; ISO-18529, 2000; Norman & Draper, 1986; Stephanidis et al., 1998). Adequacy of produced information reflects the performance of the IIS and is measured by its appropriateness and its usability. The adequacy of a result can be related to the availability of input data and to the system technological capacity. In the same way, the usability of a result depends on certain typical characteristics of the user and on the usage context of
the system. Unique characteristics of the computer-mediated information environment, information overload, and differences between individuals necessitate the understanding of information search and process models, which are specifically dedicated to computer-mediated environment for decision aid. This gives the path to move from individual users to a community of users and computer-assisted tasks to computer-mediated human activities, such as those stated in Stephanidis et al. (1999).

Focusing on a user’s characteristics and activities while interacting with an information system has been the subject of much research in the field of software development process. In addition, IIS development called for the integration of the human–computer interface (HCI) and the software engineering aspects, which gave rise to the Unified Software Development Process (Booch, Rumbaugh, & Jacobson, 1999). Rasmussen, Pejtersen, and Goodstein (1994) proposed the Cognitive System Engineering (CSE) approach, which has been improved and formalized by Hollnagel and Woods (2005), who introduced the Joint Cognitive Systems (JCS) engineering framework to “enhance the systems designer’s ability to see the big picture” (i.e., taking modeling concepts from engineering, psychology, cognitive science, information science, and computer science). Sousa and Furtado (2004) proposed a Unified Process for Interactive system, based on the well-known Rational Unified Process, to define the roles played by professionals from each field, their responsibility for creating and using in a complementary and efficient manner the different artifacts of the system, their planning of work throughout the lifecycle, and how they communicate as part of the same team.

These approaches provide the designers with efficient guidelines for coping with human errors through the design process and proposing formative models for predicting the interactive systems work behavior. They consider, however, the IIS characteristics only from the operational point of view, and little is said on how actually integrating the users’ needs, from the beginning, into the design and specification steps of the development of the IIS. The main contribution of this article is then to propose a move from the guidelines level to the definition of a formalized support, which allows the transmission throughout the various steps of the design process, of the knowledge collected with the specialists of the different involved domains for representing users’ needs. While ensuring this transfer of knowledge, the formalized support is defined so as to maintain the formalization language close to the way of expression of each professional from each field. A tool is developed accordingly to manage the integration of usage data throughout the design process of IIS.

2. RELATED WORKS

This section presents a discussion on several information systems development approaches in the scope, on one hand, of designing an IIS and, on the other hand, of ensuring the transmission of knowledge all along the process of integrating the users’ needs analysis results. It shows the lack of a unique formalized support for transmitting the different design knowledge involved in the design process. It ends by the statement of the needed requirements for developing such a support.
In the case where the IIS is a decision support system, one of its key features would be its adaptability to participants’ needs, that is, supporting a variety of decision-making processes, yet independent of any one in particular (Pereira & Quintana, 2002; Rasmussen & Vicente, 1989). Philips and Kemp (2002) proposed the use of tabular representation of use cases to describe the flow of events, and the use of user interface (UI) element clusters, which can be used as references to the UI prototype. Tabular use cases separate user and system actions, but separating users’ and systems’ actions as early as in requirements may be a barrier for future decisions in the project (Lauesen, 2001). Therefore, Lauesen (2001) suggested the use of task descriptions, which specify what the users and the system shall do, not dividing the work between them. Campos and Nunes (2004) emphasised the need for a better integration of models and tools. They presented a UI specification language bridging the gap between envisioned user behavior and concrete user interfaces. However this language may be difficult, if not impossible, to understand by designers (e.g., ergonomists) who are not from the computer domain.

Constantine and Lockwood (1999) developed their approach by focusing on describing tasks cases based on users’ intentions rather than on their actions. It offers a technological independent modeling, which conforms the users’ needs. An abstract prototype is defined using sticky notes on white board by the users, the clients, and the designers. The actual prototype is then built by software experts. Although providing the designers with flexibility with regards to the actual interactive components, this approach provides the different professionals neither with a knowledge transmission support nor with a development methodology for the articulation of its different phases. Kruchten, Ahlqvist, and Bylund (2001) proposed to include the UI Designer in the requirements workflow. The UI Designer analyzes the use case model and creates a use case storyboard. It is composed of a textual description of the interaction between the user and the system, interaction diagrams, class diagrams, usability requirements, references to the UI prototype, and a trace dependency to the use case. Then, the UI Designer designs and implements the UI prototype and obtains feedback from other project members, external usability experts, and users. In this approach the UI Designer must master software development, and no formal support is proposed for translating the textual needs description into a computable scheme. Limbourg, Vanderdonckt, Michotte, Bouillon, and Lopez-Jacquero (2004) applied their USIXML language to show how a user interface can be specified and produced at and from different, and possibly multiple, levels of abstraction while maintaining the mappings between these levels. Nevertheless, the different corresponding pieces of knowledge must be mutualized to end with the design of a system, which meets the performance criteria.

Based on the observation that the focus of CSE has changed from human and machine as distinct components to the JCS, and that it is more important to understand what a JCS does and why it does it than to explain how it does it, Hollnagel and Woods (2005) proposed a revised definition of a cognitive system as a system that can modify its behavior on the basis of experience so as to achieve specific antientropic ends. They proposed a consistent set of guidelines to achieve the design of an IIS, which permits to consider actions as a whole instead of a set of single responses,
and to improve the coupling of people with technology. The JCS approach offers a principled approach to studying human work with complex technology and provides a conceptual framework for analysis with concrete theories and methods for joint system modeling. This framework does not, however, provide the different people involved in the design process with support for expressing their requirements in their own professional language. No communication means is also available to share their knowledge for building the solution to the problem at work.

A typical design process of IIS can be split up into three steps: analysis, design, and development. The analysis step corresponds to the specifications sheets design. The design step corresponds to solution modeling. The development step corresponds to the technical achievement and the solution implementation. This article focuses on the analysis step, which is in essence an interdisciplinary task. It calls for the mobilization of several experts, methods, and tools of different domains, which are sometimes far from each other. The generic scheme of information flow, which is introduced in section 3.2, follows the JCS point of view, meaning that the user, the information system, its resources, and the context of use are considered as a whole. What is proposed then, in addition, is to take advantage of this scheme, to work out a means of communication between the professionals involved in the design process. This communication means will be supported by a formalism close to a software development language, which provides consistent support for the instantiation of the design guidelines.

The design of an IIS calls for the definition of two steps of users' information needs specifications: the system usage activity representation and the system usage context definition. The realization of these two steps relies on the description of the intervening entities, of the users activities, and of the usage contexts. Section 3 describes a general approach for structuring the definitions of these entities and their representation in a IIS specification-aid modeling tool. The application of the RAMSES approach is illustrated in section 4 with the project P@ss-ITS, which was used to support this work. Section 5 summarizes the main results of this study.

3. RAMSES: AN APPROACH FOR THE SPECIFICATION OF IIS

This section presents the different steps of the approach. RAMSES, which stands for Méthode de Recueil, d’Analyse, de Modélisation, de Spécification Et d’aide la Spécification (Data Collection, Analysis, Modeling, and Specification Method), starts from a generic representation of information flow in an IIS, which introduces its different components and their role in the system. Then, the different steps for collecting, analyzing, and representing data for the system modeling are presented. Finally, the validation step consists of defining a matching function, which outputs the solutions proposals corresponding to a given user’s needs problem.

3.1. Information Flow

Each professional actor who intervenes in the design process of an IIS generates a piece of knowledge, which contributes to the achievement of the project
at work. The RAMSES approach consists of structuring this knowledge into two flows (Figure 1). The first one moves up from the users’ needs to the specifications and represents the needs definition flow. It aims at translating the actual activity description into a conceptual representation language along a gradual abstraction process. This flow concerns all the actors and starts in the application domain by the identification of the actual users’ needs. The ergonomics specialist contribution to this step consists of collecting data by observing the activity of use of equivalent systems or contexts. Once collected, data are analyzed to generate knowledge representing the needs, the expectations and improvement recommendations including innovation suggestions. This knowledge is transmitted to the designer who must build the design models on this ground. At the end of this step, the first specifications are written and can be validate in a second flow. This flow moves down from the specifications to the needs and represents the specifications validation flow. It consists of measuring the adequacy of the designers’ specifications to the ergonomists’ recommendations. These steps are repeated until the functional specifications of a first prototype of the IIS can be set. Once implemented, the prototype will be evaluated by the ergonomist with a sample of users to update the needs reference space.

3.2. Generic Scheme of Information Flow

The functions of an IIS aims at supporting, aiding, and/or extending the users’ activity. The aim of these functions is to meet the users’ needs by outputting relevant information in the system use context (Stephanidis, 2000). The central idea to the JCS approach is that when designing an IIS, human and his environment—including the technical tool—constitute a functional unit
FIGURE 2  Scheme of information flow in an IIS.

(Hollnagel & Woods, 2005). In the case when the main purpose of the system is to provide the user with information following a user’s request, the system and the user constitute the two cognitive systems that must be joined during the design step. Information flow from and to the users can be presented as in the scheme of Figure 2 (Mouloudi & Morizet-Mahoudeaux, 2007). It shows that a user’s information needs are always transmitted to the IIS by some interface system (Media). Once transferred to the IIS, the needs must be characterized and qualified. Then a solution can be built based on available and relevant data resources. The IIS can then transmit the requested information to the user by using some adapted interface system. Needs and available information represent the use context.

The main interest of the scheme of Figure 2 is that it is a formalized framework to set the entities, which are mandatory to model the IIS functioning. Typically, designers rely on certain methods to structure their analysis for building their model. They are mainly based on development cycles management; however, these methods do not generate the data to be modeled. In other words, it is mandatory that some method makes it possible to find out in the first place what (with reference to the JCS approach) the model must represent. In the following section the RAMSES method defines the different steps of users’ needs collection: (a) gathering data from users’ needs and context, (b) representation of user data, (c) modeling of corresponding knowledge in a specific formalism, (d) specification of the system, and finally (e) evaluation of the model by a tool.

3.3. The RAMSES Method

RAMSES takes place in the preliminary analysis phase, before the design phase. The first step is defining the users’ needs, the domains experts requirements, and the available techniques for the design of the solution.

The ergonomics specialist analyzes data to formalize proposals, which meets the users’ needs and produces usage-oriented functional specifications. The functional designer can then take in charge the modeling step. Two corpuses are then available for elaborating the IIS specifications: application-domain-related data and users’ activity-related data. These two corpuses can then be modeled for generating the functional specification of the IIS and develop the first prototype for testing. A key feature of the RAMSES method lies in the last step of specification (fifth step) for evaluation, just before the development of a prototype and its validation with users. The availability of this step in RAMSES is a direct consequence of the possibility to represent users’ needs and use data in formalisms that favor
the dialogue between the designers and the ergonomics specialists. The aim of this step is to exploit the ergonomics expertise to verify the adequacy of the planned functions and their information representation modalities with the actual users’ needs. The introduction of possible specifications adjustments at this step tends to lower the number of redesign cycles. Each step of the method is formalized and equipped with specific tools. The aim of this formalization is to structure data in representations schemes, which favor their transfer from one step to the following. The following subsections present the details of each of these steps.

Data collection. This step aims at defining the user’s needs as well as the constraints linked to the IIS. The objective is to understand and characterize interaction activities of the user with the system according to its use environment. This step is realized by an ergonomist, whose task will be to specify the significant items according to the user’s point of view. In addition, a dialogue with experts of the application domain must be set to fix the limits of the technical and organizational possibilities of the system to design. To collect these data, several kinds of methods can be used: queries, interviews, group discussions, observations, and so on (Shneiderman, Plaisant, Cohen, & Jacobs, 2009). These methods are more or less close to the actual use activity. They often must be firmly adapted to the design context. For that purpose a good knowledge of the application domain is mandatory for collecting significant data. The data collection strategy is built according to an experimental planning, which is comprised of users’ sampling, definitions of the domain of query, the types of data to collect, and the tools for collecting data. One of the main properties of these methods is that they propose a strict frame for building experimental planning, users’ sampling definition, data collecting methods, and data analysis tools.

- Usage environment analysis: It consists of the description of the places where the user will access the information delivered by the system (office, vehicle, outside, etc.), the user’s aims (prescribed tasks, or tasks set by the user), and the conditions, which may influence the use modalities (rhythm, flows, temporality, etc.). Second, the available and possible media for delivering the system information (computer, mobile phone, public screens, information stand, etc.) must be listed. Media will be described according to their ergonomic characteristics (e.g., the size of the screen), and their technical characteristics (e.g., real-time information updating).

- Operating environment analysis: An information system is an information-processing application, which relies on databases and operating resources (e.g., available media and their distribution); it can also interact with other systems (e.g., to compare data, to propose degraded solutions in case of a failure of a system, etc.). Another indicator of the operating environment concerns skills procedures (in terms of instructions to follow). Effectively, for certain applications, the IIS functioning is constrained by security, reliability, or organization rules. The experts describe the functional characteristics of these resources to the ergonomists to take into account these feasibility constraints for the future system. However, the ergonomists will endeavor not
to limit the proposal to existing systems for proposing new services, which meets the users’ expectations.

- **Users’ profiles:** The definition of a representative sample of the users is mandatory to correctly define the relevant criteria of the application. The size of the sample depends on the system use environment, according, for example, to the Nielsen’s criteria (Nielsen, 1993).

- **Users’ observation:** Activity study relies on the observation of the users’ activity in actual use conditions. Observations are realized according to the operating environment just described. Two observations modalities are possible: actual context usage of an existing system and scenario-based usage. In the first mode, the user will be observed during a usual activity, whereas in the second one, the ergonomists will propose to the user a contextual scenario, which specifies particular usage modalities (Carroll, 2000; Cordes, 2001; Roth & Patterson, 2005). According to Hertzum and Jacobsen (2001), observation must be performed by different observers.

The tools suggested for describing usage and operating environment, and for setting the users’ profiles, are the classical tools for activity analysis preparation. Users’ activity analysis relies on observation, and users’ comments, which are requested to understand their underlying reasoning. Collected data must be structured to be used in the next steps. For this purpose, RAMSES proposes a grid to register activity. It is based on activity decomposition into three phases: strategy preparation, action planning, and knowledge capitalization (Figure 3).

The activity cycle can be repeated several times during the observation. If an event disturbs the course of an action plan, the user may have to do an intermediary assessment to set a new plan and put it into action. In this case, several sheets to collect the different action plans will be prepared or output if needed.

![Activity cycle diagram](Image)
**Collected data representation.** Analysis of collected data consists of extracting the specifications of the system linked to the users’ needs. It cannot be limited to the classical approach of trying to match a current request with one out of a set of previously recorded behaviors (Ricci, Arslan, Mirzadeh, & Venturini, 2002). It calls for using a formalism to represent the users’ activity. The aim of this representation is to put into light the significant elements of the users’ activity in the context of use (Booch et al., 1999). Research in the domain of human–machine interface gave rise to several formalisms of users’ tasks modeling for the specification of human–machine interfaces (Diaper & Santon, 2004). Rather than these approaches, which impose their formalism to the activity description, RAMSES proposes a representation paradigm that is built directly from the activity decomposition presented in the previous section.

The aim of activity analysis in RAMSES is to identify the information needs according to different use contexts. It is thus mandatory to take into account the dynamic of the activity in evolving contexts and with a largely variable population. An action has a meaning only if it is linked to its context occurrence and sets the general strategy progress of the user (Suchman, 1987). The collected data analysis must allow the transcription of this dynamic activity into an easy sharable format to end with an IIS specification corresponding to the targeted users’ needs.

Data are analyzed in two steps. First, data are written according to a use activity storyboard. Each step of the activity is split into sequences named **objective-action sequence**. Each user’s action, which is motivated by an information-seeking aim, constitutes a sequence. Each sequence is defined in space (usage environment description) and time (action chronology). Because each interaction of the user with information is supported by a medium, this media is named to specify the characteristics, which may have influenced the usability (or unusability) of information. The second step consists of describing the user’s plan strategy according to the **objective-action sequence**, availability of information, and adequacy of information to the needs and aims.

The Quintilian’s hexameter has been used for the first step (Valentin, Lemarchand, Mouloudi, & Morizet-Mahoudeaux, 2006). This tool is frequently referred to as quality approaches, especially in the Six Sigma method (Harry & Schroeder, 2000). It allows one to describe the circumstances of the activity: its place, its means, its way, and its time of occurrence. The aim is to tend to the completion of information description to ease its future analysis and the identification of the causes of an event.

The use of the Quintilian’s hexameter is as follows (an example is given in Table 1):

- The first level of analysis consists of the transcription of data to produce a chronological and structured story of the activity. Each observed activity is described according to a succession of significant actions.
- After setting the action (**what**) in space (**where**) and time (**when**), the observer describes its objective (**why**). Each action is analyzed to understand its actual motivations (**what for**). They can be justified by past experiences and internal
<table>
<thead>
<tr>
<th>Time</th>
<th>Where</th>
<th>What</th>
<th>Why</th>
<th>What for</th>
<th>How</th>
<th>Incident</th>
<th>Results</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>13H20</td>
<td>Terminal</td>
<td>Looking for line</td>
<td>To find the</td>
<td>Finding the</td>
<td>Looking for</td>
<td>Cannot read the map (blind</td>
<td>Looking for a phone or call the</td>
<td>Vocal media needed</td>
</tr>
<tr>
<td></td>
<td>16 bound to</td>
<td>16 bound to</td>
<td>city hall</td>
<td>city hall</td>
<td>the city hall on a</td>
<td>map</td>
<td>information center</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ST. Privé</td>
<td>ST. Privé</td>
<td></td>
<td>= Finding</td>
<td>map</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>downtown</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
representations. These motivations are used in the next steps of the design process to put into light the mechanisms, by which input data are combined with previous knowledge to build elaborated strategies. These analyses bring to light the actual motivations or objectives of each undertaken action.  
- The means used for the realization of the action (how) are then described.
- The events that may have hindered the realization of the action (incidents) are recorded.
- The consequences of the objective-action sequence—the continuation of the planned strategy or, on the contrary, a new planning due to the incident occurrence—are finally listed to register the results of the activity.

This decomposition allows one to reconstruct the logic followed by the user to reach his or her objectives. Moreover, the first analyses, which are made for identifying the actions motivations, make it possible to go beyond the elicited elements and to study the actual motivations of an action, which is often constrained by the possibilities available at its time of execution.

The second level of analysis considers the objective-action sequence from the point of view of information use. It aims at exhibiting the contextual elements, which may influence the usefulness and the usability of information. The analysis is based on the founding model of RAMSES (Figure 2):

- The user: seen from the point of view of its profile.
- The action: according to the media (and its location), which carries information.
- The elements with which the user interacts: information type, state, and content availability. The information type depends on the application domain. The information state describes the accessibility, the usefulness, and the topicality of the content for the user.
- The “Use”: according to whether the user has effectively used the information.

Analyzing each action according to information descriptors brings to light the users’ needs in the context of information use (Cordes, 2001). Each need gives rise to an information proposal or an improvement of information delivery by the ergonomist. These proposals are expressed in terms of resources (Figure 2), which have been identified during the analysis of the application domain.

At the end of this step the ergonomists have a qualified knowledge corpus. This knowledge describes information use activity in critical situation describing a needs to satisfy. At the end of these two analyses the ergonomists can propose use-oriented functional specifications of the IIS. These specifications can be updated according to the technical possibilities and constraints of the environment.

---

1It is the personal representations of the source and the addressee of a message, which let emerge its meaning (Wiener, 1948).
**System modeling.** The aim of this step is to build the formal model of the knowledge acquired in the previous experimental step and to represent it with a structured formalism for its implementing by a logic model.

The move from needs representation to a formal model consists of finding the paradigm and its corresponding concepts to represent the entities (and their relationships), which convey a meaningful representation of the problem. The choice of a formalism depends, on one hand, on the ability of its concepts to maintain the integrity of the collected data and, on the other hand, on the language, which will be used for the system implementing.

The formalism and the corresponding data representation language, which are appropriate to software engineering, must be mastered by the software designer and meaningful for the ergonomist who analyzed the users’ needs. They must permit to link the contextual users’ needs with the formalized information resources data while preserving the richness of the rigorous data collection and analysis method presented in the previous section.

The object-oriented (OO) approach provides a powerful and effective environment for modeling and building complex systems. It supports a variety of techniques for analyzing, designing, and implementing flexible and robust real-world systems, providing benefits such as encapsulation, polymorphism, inheritance, and reusability (Booch et al., 1999; Jacobson, 1992). It permits the building of the conceptual representation of data issued from the real world and allows their implementing (Détienne, 1994). They advocate for translating the experimental data according to an ontological description of the real-world entities and their relationships.

The Unified Modeling Language (UML) was adopted as a standard for OO modeling by the Object Management Group in 1997. UML allows modelers to represent multiple independent perspectives or views of a system using a variety of graphical diagrams, such as the use case diagram, class diagram, state diagram, sequence diagram, and collaboration diagram. Because the RAMSES approach relies on use case studies, UML appears to be a particularly adapted language. Moreover, a use case model acts as an effective tool for communication between developers and end users (Agarwal & Sinha, 2003). Whereas a class diagram portrays the static structure of a system, a state diagram captures its dynamic or behavioral nature, in terms of state transitions.

There are two types of interaction diagrams: sequence diagrams and collaboration diagrams. Sequence diagrams show the explicit sequencing of messages; collaboration diagrams show the relationship among objects. Both are useful for developing a dynamic model of a system in terms of object interactions. Interaction diagram is used to show the pattern of interactions among objects for a particular use case. Objects communicate with one another by sending messages. Although further work still remains to merge the achievement of HCI research into mainstream UML-based software design methodologies (Palanque & Bastide, 2003), OO developers perceive the UML approach as easy and efficient to use, especially for use case diagrams and state diagrams (Agarwal & Sinha, 2003). Developers with more experience in OO analysis and design techniques perceived class diagrams and interaction diagrams also as being easy to use (Agarwal & Sinha, 2003). In the case of IIS, some specific approaches have already shown ways
of improvement. For example, da Silva and Paton (2003) proposed an extension of UML to provide greater support for user interface design. UML permits one to model tasks using extended activity diagrams rather than by incorporating a completely new notation into UML. UML also addresses the relationships between use cases, tasks, and views and thoroughly addresses the relationship between tasks and the data on which they act. In their survey conducted on a method that integrates the use of UML and a formal notation, Razali, Snook, Poppleton and Garratt (2008) showed that the method is accessible to users when the principles and roles of each notation are obvious and well understood and when there is strong support from the environment.

The RAMSES modeling process is set along two axes: a static modeling and a dynamic modeling. The static modeling describes the use and operating environment of the IIS. The static classes diagrams, which represent the description of the problem at work, are built based on the information flow presented in Figure 2. Each element of the environment (place, media, operator, etc.) is represented as a class. Activity analysis permitted to describe the components of these elements, which will appear as the attributes and methods of these classes. More precisely, the ergonomic analysis of the previous step permitted to define users’ profiles, the media, and the resources in their context of use. Classes are then linked by associations, which describe their interaction in the process of production and distribution of information between the system and the users. The specification of the system at this step is to describe the expected functions of the system after the analysis of the users’ needs. They are also the results of the ergonomist observation and analysis of the previous step. An example of static modeling is presented in the Domain Modeling section.

The second axe of modeling concerns the dynamic aspects of users’ activity. This model makes explicit the interactions of the user with the entities described in the static model. It describes the observed users’ activities, based on the action-objectives sequences. Each action-objective sequence represents a call to some object. Thus it is a class method, which can induce a change of state of the class.

The system modeling is realized by a software designer. However, it is directly built from the specifications issued from the ergonomic activity analysis of the previous step, which have been transcribed in a structured and formalized format. This permits transfer of the users’ needs, which have been qualified by the ergonomist, from their semantic description to a computer implementable description. An example of dynamic modeling is presented in the Activity Modeling section.

**Users’ needs matching function.** A users’ information needs matching function is defined thanks to the previous modeling. It links the usage activity to the resources of an IIS. Information needs appear whenever the user looks for a solution to a given problem corresponding to a given contextual use. Once identified, each need brings out the building of a solution. In other words, a needs match \( (N_M) \) is a projection of a situation defined by an event \( E \) and a context \( C \) into the solution space, expressed in terms of data resources solution proposal \( P_R \). The matching function can be formalized as follows:
The pair \((E, C)\) describes a usage situation, which brings out an information needs. Each situation is linked to one or several proposals, which have been exhibited by the ergonomist. This solution is expressed in terms of the data resources, which are needed for building it.

A solution proposal may be linked to two or more distinct usage situations.

\[
\exists (E_1, C_1), \exists (E_2, C_2), (E_1, C_1) \neq (E_2, C_2) \Rightarrow N_M(E_1, C_1) = N_M(E_2, C_2)
\]

Each solution proposal corresponds to at least one usage situation.

\[
\forall P_R, \exists (E, C), N_M(E, C) = P_R
\]

A specification-aiding tool for the designer can be implemented based on this function and to the modeling presented in the previous sections. The model permits one to exhibit a view of the problem in the shape of linked classes and objects. Implementing these classes according to the needs satisfaction function permits one to propose a tool for managing our knowledge of the users’ needs. It makes it possible to visualize the needs and the corresponding usage-oriented functional recommendations, stated by the ergonomist. These recommendations will be expressed in terms of entities of the domain model. The designer can then update the system architecture according to these new needs, which were not necessarily taken into account in the first steps of the IIS design.

At the end of the process of needs identification and modeling, and according to the technical specifications, the designer has a model of the system or of the prototype. A dialogue can then begin between the designer and the ergonomist based on this model to confirm its adequacy with the users’ needs. After completion of the discussion, the designer can build a first prototype, which the ergonomists will assess with a sample of users.

4. APPLICATION TO THE SPECIFICATION OF A TRAVELER INFORMATION SYSTEM

The RAMSES approach has been used in the case of a traveler information system (TIS) design project, \(P@ss-ITS\). The \(P@ss-ITS\) project was funded by the PREDIM program and was developed in cooperation with the Alstom Company, the Transdev Company, and the INRETS Research Laboratory (Institut National de Recherche sur les Transports et leur Sécurité). The aim of this project was to evaluate the advantages of providing the users of a multimode transportation system with real-time information, especially in the case of disturbance (roadwork, accident, scheduled or not-scheduled event). This project was tested on a middle-sized urban area of about 300,000 inhabitants. The area has a bus network, a streetcar network, and a central train station. The system must provide updated and
relevant information to travelers built from a central information system, which centralizes all information sources of the urban area (Figure 4). It must also provide the users with relevant answers to their requests (itinerary, travel schedule, travel modification, search, etc.).

Information sources are as follows:

- Operation aiding systems: They are computer architectures linked to the operation centers of the different transportation modes. They assist the operators in the management of the networks.
- Control station: They provide real-time information concerning the traffic in the area.
- Interactive media: These interface systems provide users with information and send users’ information (request, etc.) back to the central system. The supported dialogue is invaluable in building a personalized information expression.

These data are managed by the system to express the relevant real-time message, which will let the user decide how to travel, even in the case of a disturbance of the transportation network.

The following sections present the use of the RAMSES method in the case of the P@ss-ITS project.
4.1. Data Collection

The variability of the population, use contexts, and objectives in the scope of public transportation system use must be taken into account for defining the population samples, the contexts, and the travels. Data have been collected by an ergonomic observation of the activity during actual travels. Tools of chronological transcription have been defined to record the observed data and contexts. Then, each step of the travels has been analyzed according to the action-objective sequence approach defined in the method.

Users sampling. A representative sample of the users’ variability has been built based on a preliminary sociodemographic study (the detailed description of the sample building study can be found in Mouloudi, Lemarchand, Valentin, & Morizet-Mahoudeaux, 2005). The selection criteria have been defined in accordance with the domain experts. Some criteria concerned the impact of the motivation, time, and constraints on the travel; for instance: people who commute have frequent and regularly scheduled travels from home to working place, people traveling for their leisure can choose to travel at midday, elderly people may have problems moving easily.

Other aspects may change information needs; for example: the knowledge of the transportation network may change the needs in terms of map or timetable, disabilities may imply specific information modes (e.g., people who are blind) or specific equipment. Students represent an often traveling population, and strangers to the area have longer travels.

The study is based on a sample of 36 people representing the population (students, older people, workers, strangers, etc.). These profiles have been mixed to enrich the representativeness of the sample; for example: season ticket holder and worker, older person and stranger, student and person with a disability. The sample contained at least 6 persons for each criteria, which corresponds to a representativeness of at least 80% of the corresponding profiles (Nielsen, 1993; Valentin et al., 2006).

Contexts sampling. The urban area consisted of a midsize town and 22 surroundings villages, and the transportation network consisted of one streetcar line and 30 bus lines. Lines’ time tables varied with the transport mode and the localization. The streetcar line and some bus lines have a high frequency, whereas some others have variable frequencies depending on the day or the time of day. The network has many connection points; some of which gather many lines (12 lines at the central train station, 8 lines at Léon Blum station), whereas the average is 2 or 3 lines.

A sample of contexts for the observation phase has been defined based on the criteria defining the network (transportation mode, localization, frequency, type of connection). They are, for example, high-frequency streetcar line, regular-frequency bus lines, low-frequency bus lines, big network connection station, small station, and so on.
Identification of the usage situations. Two different observation modes were defined for the study. In the first one, the user traveled in the nominal usage environment, and the use of the media and information provided by the TIS was observed. The usage context was known by the observer. The observer collected data on the events, which appeared during the interactions. The second observation mode was the matter of updating usage strategies, which could be specific on certain usage situations. The observer simulated perturbation events, such as inaccessible station or impossible connection, and observed the user’s activity corresponding to the new situation. Many events were thus introduced and taken into account in the contexts variability. Each user was observed during the travels for about 2 hr.

Actual situation observation. The ergonomic approach relies on the activity analysis of actual situation and puts the observation at the center of the method (Nardi, 1995). Specific information collection sheets have been defined for this study. They allowed a systematic record of elaborated action; seek, used, or expected information; incidents and events; and so on. The user had to also explain the strategy and choices, which were also recorded. A discussion was also set at the end of the travel to summarize the encountered events.

Chronological transcription. Data have been transcribed according to the method described in the previous section: The chronological representations of the travels have been built with the Quintilian’s hexameter. Then the travels have been split into action-objective sequences and classified along the chronological axis of the travel. Each sequence was defined by the time and the place of the action, its aim, and the used modality. Each action was then analyzed to extract its characteristics. They can be the result of previous travel experiences or individual representations. The actual motivations and objectives of the user could then be exhibited.

Table 1 shows an extract of experimental data transcription. The column Incident lists all the events, which could disturb or hinder the user’s action. The Results column describes the consequences of the action-objective sequence. They can be, for example, the continuation of the moving strategy or a change of direction according to the incident occurrence. The last column, Comments, has been added to include the comments of the observer about the context of the described sequence.

Information use processing. The second level of processing consisted in qualifying each action-objective by descriptors of the use and usability of information in the context of the activity. In addition to the items appearing in the preceding processing, each sequence has been placed in the context of a strategic step of the user’s travel (scheduling, realization, knowledge capitalization; see Figure 3), the concerned information media, the state of information at the time of the action (up to date, wrong, missing, requested, not available, useless, etc.), its use by the user (used or not, misleading, etc.), the level of expertise of the user according to the foreseen action, the aim of information use. An example is given in Table 2.
Table 2: An Extract of the Information Use Analysis Table

<table>
<thead>
<tr>
<th>Ref</th>
<th>Ch</th>
<th>Place</th>
<th>N</th>
<th>Media</th>
<th>State</th>
<th>Use</th>
<th>Level</th>
<th>Objectives / Actions</th>
<th>Proposals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A3</td>
<td>89</td>
<td>Terminal C-A</td>
<td>1</td>
<td>District map</td>
<td>Needed</td>
<td>I</td>
<td>1</td>
<td>Looks for Line 16 bound to St Privé</td>
<td>Indicating stations on the district maps</td>
</tr>
<tr>
<td>A3</td>
<td>90</td>
<td>Terminal C-P</td>
<td>1</td>
<td>Network map</td>
<td>Missing</td>
<td>I</td>
<td>1</td>
<td>Impossible to check the itinerary</td>
<td>Systematization of network maps presence</td>
</tr>
<tr>
<td>A3</td>
<td>91</td>
<td>Terminal C-S</td>
<td>1</td>
<td>Display</td>
<td>Missing</td>
<td>N</td>
<td>1</td>
<td>The user leaves the station</td>
<td>Ensure information redundancy on different media</td>
</tr>
<tr>
<td>A3</td>
<td>92</td>
<td>Terminal C-S</td>
<td>1</td>
<td>Ticket vending machine</td>
<td>Missing</td>
<td>N</td>
<td>1</td>
<td>The user leaves the station</td>
<td>Check for the media state</td>
</tr>
</tbody>
</table>
This table is helpful in drawing conclusions at several levels. The first level concerns the actual use of information media in their different places of travel. It brings to light the consistency of the proposed information according to the user’s strategy. Second, the information needs at each step of the travel can be checked. For this, all users’ data are merged and the actions are grouped according to chronological criterions (planning, arrival, exit) or geographical criterions (the place). Specific needs for some users’ categories can also be exhibited.

4.2. Data Modeling

The next step was to integrate these results in the design process to build the users’ needs matching function (1) for its evaluation. Thus it was necessary to translate these data into a language for implementation. According to RAMSES, the modeling was developed along two axes: a static model of the domain and a dynamic model of the activity. Then the model was translated into UML notation to obtain an easier interpretation for its implementation.

**Domain modeling.** The static model of the domain gathers all the entities, which interact with the information production. The knowledge necessary to describe the contextual use of the TIS was deduced from the analysis of users’ activity. The corresponding entities and their relationships have been defined following the modeling process. The static model (Figure 5) has been built based on the formal scheme of information flow presented in Figure 2. The static model exhibits the entities, which interfere in this flow. Each entity is represented in the shape of a class diagram.

**Activity modeling.** The dynamic activity of users has been built based on use cases diagrams. With the UML language, a use case diagram represents the set of all configurations, which characterise the system behavior. It describes the structure of the functions, which are necessary for users. The set of use cases is aimed at describing exhaustively the functional requirements of the system from the user’s point of view. A synthetic view of the model is presented in Figure 6.

The use cases have been observed for all the traveling users for different steps of their travels and different order of occurrences. Then, each use case was described by strategic diagrams. These diagrams aim at exhibiting the particular strategy of each user to reach his or her objective. Each strategy uses different resources and involves different needs. The diagrams have been built based on the action-objective sequences decomposition of users’ activity. Packages of sequences corresponding to typical situations of information use have been built accordingly. The corresponding data packages have then been analyzed to distinguish the objectives, which are common to several users from those, which are specific to some user’s profile.

Two kinds of UML diagrams are available to represent strategies: sequence diagrams and activity diagrams. Activity diagrams can represent the different states
of the objects along the progress of the strategy. However, activity diagrams do not show explicit time parameter. On the other hand, sequence diagrams, which are based on the chronological message exchanges between objects and the user, seemed in this case to be more appropriate. In addition, sequence diagrams were also appropriate to represent the variability of the strategy used by travelers to reach their objectives.

4.3. Evaluation

A class-based model of the application domain and of the use cases diagrams comprised of activities diagrams and sequence diagrams was thus available from the UML modeling. Java classes have been built based on this model. They were
implemented from the experimental data stored in the tables such as Table 2. It was then possible to develop a tool for evaluating the adequacy of solution proposals to a given user’s information needs. This tool has been built based on the evaluation function presented in the users’ Needs Matching Function section.

A significant part of the UML class-based model has been translated into the Java programming language. The UML notation facilitated the translation from the model into the language code. The specification-aid modeling tool has been developed according to the Model View Controller design pattern. The designer selects the value of criteria for describing a usage situation (context + events), and the tool outputs the state of the resources needed to provide the user with the right information according to the context. The output is computed by using the evaluation function.

One limitation to the generalization of the methodology is to define the different actors’ profiles and background. All of them must work in a collaborative way. Indeed RAMSES provides the framework and tools to represent information, that is, a support to send the messages from one step to the other of the design process. However, it is necessary to work on providing the actors with the ability to understand information generated at the preceding step of the process, that is, to receive and understand the messages.

5. CONCLUSIONS AND FUTURE WORK

In this article, a methodological frame and the associated tools to promote the transmission of data representing the users’ needs from the analysis step to the design step within the development process of an IIS has been presented. This approach has been applied to the design of a traveler information system. It has been developed so as to guarantee the respect of the interdisciplinary dimensions of this process. This study and its application to a real case try to demonstrate that a novel use of the ergonomic concepts and tools, of the knowledge representation paradigms, and of the software engineering modeling methods, can ensure the transmission of knowledge along the different steps of the design process. The difference of understanding of the problem by the different actors of the design process can thus be lowered and promote a better integration of the users’ needs in the design of IIS. This approach can also be used for evaluating a specification model and its evolution along the life cycles of the design process.

The next step of this study will be to validate its application in the frame of a design project allowing its complete achievement. So, the outcomes of specifications validation, provided by an evaluation function, could be tested before the implementation of prototypes. It will consist of assessing its contribution in terms of needs satisfaction and interprofessional communication and in measuring the extent to which the usage specifications have been transmitted up to the technical specifications. Activity analysis can be used in all the phases of the design loops, to assess model or prototypes, and the RAMSES method can be an efficient support for each of these.

A further improvement of RAMSES could be to add iteration loops. Following the user-centered approach, this would consist of analyzing the actual use by the
sample of users of the new services provided by prototype. This experiment would be built according to the RAMSES method: definition of the sample, observation, collection of data, data analysis, and evaluation.

Another way to go forward with RAMSES would be to study its application to a completely different design process than UP, such as SCRUM, Agile, and eXtreme Programming. The teams in this type of project are mainly centered on the technique while one important factor of success of the project is the value delivered to the end user. In addition, the development cycles are considerably shortened. The combination of these factors makes even more critical the necessity of the representation and the transmission of user needs during the design process.

REFERENCES


