Improving Interactive Systems Usability using Formal Description Techniques: Application to HealthCare

Philippe Palanque, Sandra Basnyat, David Navarre

Improving Interactive Systems Usability Using Formal Description Techniques: Application to HealthCare

Philippe Palanque, Sandra Basnyat, and David Navarre

LIIHS-IRIT, Université Paul Sabatier – Toulouse III
118 route de Narbonne, 31062, Toulouse Cedex 4
{palanque,basnyat,navarre}@irit.fr

Abstract. In this paper we argue that the formal analysis of an interactive medical system can improve their usability evaluation such that potential erroneous interactions are identified and improvements can be recommended. Typically usability evaluations are carried out on the interface part of a system by human-computer interaction/ergonomic experts with or without end users. Here we suggest that formal specification of the behavior of the system supported by mathematical analysis and reasoning techniques can improve usability evaluations by proving usability properties. We present our approach highlighting that formal description techniques can support in a consistent way usability evaluation, contextual help and incident and accident analysis. This approach is presented on a wireless patient monitoring system for which adverse event (including fatalities) reports are publicly available from the US Food and Drug Administration (FDA) Manufacturer and User Facility Device Experience (MAUDE) database.

Keywords: Human–Computer Interaction, Incident and Accident Investigation, Formal Description Techniques, Medical Informatics, Patient Monitoring.

1 Introduction

The advances of healthcare technology have brought the field of medicine to a new level of scientific and social sophistication. They have laid the path to exponential growth of the number of successful diagnoses, treatments and the saving of lives. On the hand, technology has transformed the dynamics of the healthcare process in ways which increase the distribution & cooperation of tasks among individuals, locations and automated systems. Thus, technology has become the backbone of the healthcare process. However, it is usually the healthcare professionals who are held responsible for the failures of technology when it comes to adverse events [8]. It has been estimated that approximately 850,000 adverse events occur within the UK National Health Service (NHS) each year [26]. A similar study in the United States arrived at an annual estimate of 45,000-100,000 fatalities [17]. While functionality of medical technology goes beyond imaginable, the safety and reliability aspects have lagged in comparison with the attention they receive in other safety-critical industries. The discussion of human factors in medicine has centered on either compliance with government standards or on analyzing accidents, a posteriori.

In this paper we are interested in medical systems that offer a user interface and require operators interaction while functioning. Due to their safety-critical nature there is a need to assess the reliability of the entire system including the operators. The computer-based part of such systems is quite basic (with respect to other more challenging safety-critical systems such as command and control systems (cockpits, Air Traffic Management, …)) and thus current approaches in the field of systems engineering provide validated and applicable methods to ensure their correct functioning[1]. Things are more complicated as far as the user interface and operators are concerned. Human-computer interaction problems can occur because of numerous poor design decisions relating to the user interface (UA). For example, poor choice of color, a mismatch between the designer's conceptual model and the user's mental model or insufficient system feedback to allow the user to understand the current state of the system which is known as mode confusion. Mode confusion refers to a situation in which a technical system can behave differently from the user's expectation [9]. Operator assessment is even harder to perform due to the autonomous nature, independence and vulnerability to environmental factors like stress, workload.

Lack of usability has been proved to be an important source of errors and mistakes performed by users. Faced with poor user interfaces (designed from a non-user centered point of view) users are prone to create alternative ways of using the applications thus causing hazardous situations. In the worst cases, lack of usability may lead users to refuse to use the applications, potentially resulting in financial loss for companies with respect to the need for redesign and possibly additional training. For detecting and preventing usability problems it is important to focus the design process from the point of view of people who will use the final applications. The technique called User-Centered Design (UCD) [30] is indeed the most efficient for covering user requirements and for detecting usability problems of user interfaces.

Usability evaluation, both formative and summative, can improve these kinds of issues by identifying interaction problems. However, after the design iteration, another round of usability evaluation is required in order to verify whether there is any improvement. This can be costly in terms of time and resources. A further way to improve design is to apply interface design criteria and guidelines during the design process. Ergonomic criteria have been proved to increase the evaluation performance of experts [6][1]. Safety critical systems have been for a long time the application domain of choice of formal description techniques (FDT). In such systems, where human life may be at stake, FDTs are a means for achieving the required level of reliability, avoiding redundancy or inconsistency in models and to support testing activities. In this paper, we will illustrate our approach using a FDT based on Petri nets. In this paper, we show that FDTs can be used to support a selection of usability related issues including:

- Formative evaluation
- Summative evaluation
- Contextual help for end users
- Investigation of incidents and accidents by providing formal descriptions of the behavior of the system

[1] As pointed out in 14 and 15 there is an increasing integration of new technologies in medical application that will raise new issues in their reliability assessment.

The next section presents an overview of usability evaluation, including formative and summative evaluation and presents several ergonomic/usability criteria and guidelines for interface design including some that are targeted at touch screen interfaces. Section 3 provides an informal presentation of formal description techniques and the Interactive Cooperative Objects (ICOs) formalism we will use in this paper, before presenting the case study in section 4. Section 5 illustrates the approach by discussing ways in which formal specification of the behavior of the system using ICOs and its environment (Petshop) can improve formative and summative evaluation, provide contextual help for end users and support the investigation of incidents and accidents. We then conclude and provide suggestions for future work.

2 Usability Aspects for Interactive Systems

Usability addresses the relationship between systems, system interfaces, and their end users. Interactive systems, or systems requiring humans in the loop of control must be designed from a UCD approach in order for the intended users to accomplish their tasks efficiently, effectively and as expected by the user. A system with poor usability can cost time and effort, and can greatly determine the success or failure of a system (See [13] for guide on usability engineering and [14] for interface design for medical applications). In this section, we firstly present a brief overview of usability testing, distinguishing between the two main forms of testing, formative and summative (2.1). We then discuss usability criteria and guidelines as a means of improving usability (2.2). This is extended in the third section (2.3) by describing guidelines for a specific kind of interface, touch screens. The fourth section (2.4) describes the importance of providing contextual helps as a means of improving reliability while the final section (2.5) introduces a slightly different dimension in which we discuss the ways in which formal description techniques (FDTs) can assist in improving the design of safety-critical interactive systems, such as medical applications, by helping to avoid incidents and accidents.

2.1 Usability Testing

The usability of a system can be tested and evaluated. Typically, the testing serves as either formative or summative evaluation.

Formative tests are carried out during the development of a product in order to mould or improve the product. Such testing does not necessarily have to be performed within a lab environment. Heuristic Evaluation (HE) [27] is an established formative Usability Inspection Method (UIM) in the field of HCI. The analyst, who must have knowledge in HCI, follows a set of guidelines to analyze the user interface. The technique is inexpensive, easy to learn and can help predict usability problems early in the design phase if prototypes are available. Examples of HEs include [35], [29] and [6]. While this paper focuses on presenting how formal description techniques (FDTs) can support and improve the usability of interactive safety-critical systems, in [8] we have shown how they can support usability testing.

Summative tests in contrast are performed at the end of the development in order to validate the usability of a system. This is typically performed in a lab environment and involves statistical analysis such as time to perform a task.

2.2 Usability Criteria and Guidelines

Heuristic evaluation is a common usability inspection method achieved by performing a systematic inspection of a user interface (UI) design for usability. The goal of heuristic evaluation is to find the usability problems in the design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics") [29]. Bastien and Scapin [6] evaluate the usefulness of a set of ergonomic criteria for the evaluation of a human-computer interface with results indicating that usability specialists benefit from their use. Since their early publications concerning ergonomic guidelines for interface design to improve usability [36], Neilsen and Molich coined the term "heuristic evaluation" [27]. Since the original publication of heuristics, Nielsen refined them to derive a set of heuristics with maximum explanatory power. The following lists present Bastien and Scapin's [6] and Nielsen's guidelines [28].

Table 1. Criteria for interface design to improve usability

Ergonomic criteria – from Table 1 in Bastien and Scapin [6]		Nielsen's Heuristics guidelines [28].	
1	Guidance	1	Visibility of system status
2	User workload	2	Match between system and the real world
3	User explicit control	3	User control and freedom
4	Adaptability	4	Consistency and standards
5	Error management	5	Error prevention
6	Consistency	6	Recognition rather than recall
7	Significance of codes	7	Flexibility and efficiency of use
8	Compatibility	8	Aesthetic and minimalist design
		9	Help users recognize, diagnose, and recover from errors
		10	Help and documentation

Based on the ergonomic criteria set out by Bastien and Scapin, they further define guidelines relating to each criterion. The following is an example of the "user explicit control" criteria. The guideline is called "user control" (see Table 2 in [6])

1 Allow users to pace their data entry, rather than having the pace being controlled by the computer processing or by external events
2 The cursor should not be automatically moved without users' control (Except for stable and well known procedures such as form-filling)
3 Users should have control over screen pages

4 Allow users to interrupt or cancel a current transaction or process
5 Provide a ‘cancel’ option that will have the effect of erasing any changes just made by the user and restoring the current display to its previous version.

We have already shown how formal description techniques can support systematic assessment of user interface guidelines in [33] by relating using explicit states representations as a link between guidelines and behavioral description of interactive applications.

The usability guidelines and criteria mentioned so far in this section relate to UIs in general. However, as we will see in our case study, the interaction between human and system with medical applications is migrating towards touch-screen interfaces. The following subsection presents guidelines relating specifically to these kinds of devices.

2.3 Guidelines for Specific Devices: The Case of Touch Screen

Tactile interfaces have, in addition to the above mentioned usability criteria and guidelines, their own interface design challenges. For example, the use of a touch screen of an outdoor cash machine in winter by a user wearing gloves may not be effective. Though standards for principles and recommendations as well as specifications such as ISO13406, ISO 14915, ISO 18789 exist for interface and interaction, they do not specifically target tactile interfaces for neither typical walk-up-and-use interfaces, nor domain specific interfaces such as those for the healthcare domain.

In a recent conference called Guidelines on Tactile and Haptic Interactions (GOTHI), Fourney and Carter [11] review existing international standards on tactile/haptic interactions and provides a preliminary collection of draft tactile/haptic interactions guidelines based on available guidance. The guidelines are categorized under the following headings:

- Tactile/haptic inputs, outputs, and/or combinations
- Tactile/haptic encoding of information
- Content specific Encoding
- User Individualization of Tactile / Haptic Interfaces

We provide below a selection of the guidelines proposed Fourney and Carter [11] that particularly relate to the research and case study presented in this paper, that is the improvement of usability of a safety-critical interactive touch screen system. See [11] for the complete list of guidelines.

- Provide navigation information
- Provide undo or confirm functionality
- Guidance on combinations with other modalities
- Make tactile messages self descriptive
- Mimic the real world
- Use of apparent location
- Keep apparent location stable
- Provide exploring strategies
- Use size and spacing of controls to avoid accidental activation
- Avoid simultaneous activation of two or more controls

[1] has also studied usability of touch screen interfaces and suggests that menus and buttons that are often selected should be located at the bottom of the interface rather than the top to avoid covering the screen (and in the case of a safety-critical application, vital information) with one's arm while accessing the menus. Additional references to research on guidelines for touch screens can be found [15] and [16].

Furthermore, critical interfaces, such as those used in to monitor patients in the healthcare domain provide vital information. This means, if selecting an option within the application forces a new window to cover the initial screen, there is a risk that it will be unintentionally missed during that short period of time (see Fig. 1). A lot of work has been done on the visibility of a workspace, with techniques designed to provide contextual awareness at all times even while the user focuses on one aspect of the interface (see [21]).

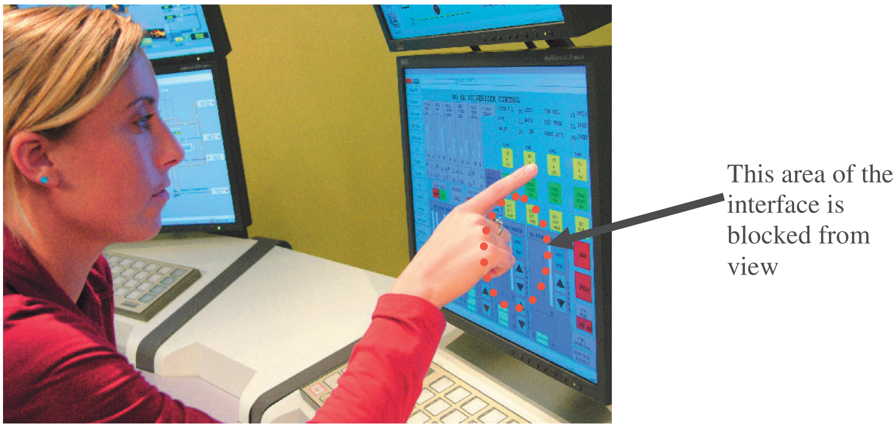

Fig. 1. Interaction with a touchscreen

Coutaz et al, [10] describe "CoMedi", a media space prototype that addresses the problem of discontinuity and privacy in an original way. It brings together techniques that improve usability suggested over recent years. The graphical user interface of CoMedi is structured into three functional parts: at the top, a menu bar for non frequent tasks (see Fig. 2 and Fig. 3). In the center, a porthole that supports group awareness. At the bottom, a control panel for frequent tasks. The "The fisheye porthole" supports group awareness. Though the example in the paper is of a collaborative tool in a research laboratory, the way in which the fisheye porthole is designed could be useful in a medical environment. The porthole may have the shape of an amphitheatre where every slot is of equal size. When the fisheye feature is on, selecting a slot, using either the mouse or a spoken command, provokes an animated distortion of the porthole that brings the selected slot into the centre with progressive enlargement [10].

The idea of context loss while zooming or focusing on a particular aspect of the interface has been argued and researched by Bob Spence with his database navigation approach [37], Furnas, with his generalized fisheye views [12]and Mackinlay et al.,

Fig. 2. The graphical user interface of CoMedi

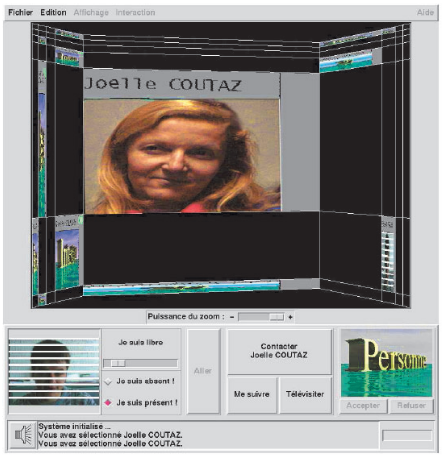

Fig. 3. The UI of CoMedi with Porthole when the fisheye view is activated

[20] with their perspective wall his perspective wall approach and by John Lamping with his hyperbolic browser [18].

2.4 Importance of Contextual Help

Contextual help is an important feature that interactive systems should include. It refers to the available of help at any point in time relative to the given state of the system. For example, if a user expects an option or a button to be available and it is not (i.e. a grayed out button in a menu), the contextual help would indicate why this option is not available at that moment. Formal description techniques (FDTs) provide a way to have a complete and unambiguous representation of all states of the system and state changes as well as a means for reasoning about these states and state changes. However, without formal methods fully describing the behavior of the system, providing this kind of advice is very difficult. By simply analyzing a user interface, it is almost impossible (except with a very basic system) to know exactly what state the system is in. In order for the formal description of the system to be directly usable for the end user (i.e. the operator), it would certainly need additional annotations. It is unlikely that medical personnel would be able to read such a model without training and it is not useful to train them in such a way. The main point is that the formal model supports designers and developers' activities by for instance identifying why users are experiencing a problem and how to overcome it. Furthermore, analyzing lines of code to determine the system state is extremely cumbersome and providing developers with an abstract view of the system can support them. We show in section 5.2 that a formal graphical description of the behavior of the system can provide information on the current state, why the option/button is not available and what must be performed by the user for that option to be available. More information on how such contextual help can be supported by FDT can be found in [34].

2.5 Avoidance of Incidents and Accidents

The heading incident and accident investigation can be misleading in a usability-related article, however we believe and described in the introduction, that poor usability of a system can potentially lead to an adverse event.

In terms of supporting their investigation, we have devised an approach exploiting two complementary techniques for the design of safety-critical interactive systems. More precisely we address the issue of system redesign after the occurrence of an incident or accident. The techniques used are Events and Causal Factors analysis, used in accident investigation to identify the path of events and contributing factors leading to an accident, and secondly, Marking Graphs, an analysis technique available after formally modeling the interactive system, using Petri nets for example. Marking graphs are used to systematically explore all possible scenarios leading to the accident based on a formal system model. The formal description techniques (FDTs) allow us to exploit and represent all system states. The techniques serve two purposes. Firstly, it is intended to ensure that the current system model accurately models the sequence of events that led to the accident according to information provided by Events and Causal Factors Analysis. Secondly, our approach can be used to reveal further scenarios that could eventually lead to similar adverse outcomes. The results can in turn be used to modify the system model such that the same accident is unlikely to recur. This is achieved by identifying system states that are undesirable and may want to be avoided. By identifying the state changes leading to such undesirable states, it is possible to adapt the system model to make these states impossible by making the paths (combination of state changes leading to that state) impossible. This is supported by simulating the identified scenarios on the adapted system model. This will be exemplified in section 5.3, on a simple example using the PatientNet case study. The interested reader can see a complete example in [4].

3 Formal Specification of Interactive Systems

The previous sections have described usability evaluation techniques, usability guidelines targeted specifically at touch screen interfaces as well as relating usability to incident and accidents. In this section, we present a formal description technique (FDT) for describing the behavior of interactive systems. Most FDT could provide the kind of support for usability we are describing but we present rapidly here the FDT ICOs (Interactive Cooperative Objects) dedicated to the specification of interactive systems based on Petri nets that we will use for the case study in section 4.

Formal verification of models can only be achieved if the formalism is based on mathematical concepts. This allows reasoning on the system models in addition to empirical testing once the system has been implemented. Verification of models has economical advantages for resolving problems in terms of consumption of resources. Further advantages include the possibility to locate an error (though perhaps not the exact causes for the error).

One of the aims of the formal analysis is to prove that there is no flaw in the models. Using the ICO formalism for describing the models, the analysis is done by using the mathematical tools provided by the Petri net theory. Using those tools, one can prove general properties about the model (such as absence of deadlock) or semantic domain related properties (i.e. the model cannot describe impossible behavior such as the light is on and off at the same time).

Modeling systems in a formal way helps to deal with issues such as complexity, helps to avoid the need for a human observer to check the models and to write code. It allows us to reason about the models via verification and validation and also to meet three basic requirements notably: reliability (generic and specific properties), efficiency (performance of the system, the user and the two systems together (user and system) and finally to address usability issues.

3.1 ICOs as an Concrete Example of Formal Description Techniques

The aim of this section is to present the main features of the Interactive Cooperative Objects (ICO) formalism that we have defined and is dedicated to the formal description of interactive systems. We encourage the interested reader to look at [23] for a complete presentation of this formal description technique as we only present here the part of the notation related to the behavioral description of systems. This behavioral description can be completed with other aspects directly related with the user interface that are not presented here for space reasons.

Interactive Cooperative Objects Formalism (ICO)

ICOs are dedicated to the modeling and the implementation of event-driven interfaces, using several communicating objects to model the system, where both behavior of objects and communication protocol between objects are described by the Petri net dialect called Cooperative Objects (CO). In the ICO formalism, an object is an entity featuring four components: a cooperative object which describes the behavior of the object, a presentation part (i.e. the graphical interface), and two functions (the activation function and the rendering function) which make the link between the cooperative object and the presentation part.

Cooperative Object: Using the Cooperative Object formalism, ICO provides links between user events from the presentation part and event handlers from the Cooperative Objects, links between user event availability and event handler availability and links between state in the Cooperative Objects changes and rendering.

Presentation part: The presentation of an object states its external appearance. This presentation is a structured set of widgets organized in a set of windows. Each widget may be a way to interact with the interactive system (user → system interaction) and/or a way to display information from this interactive system (system → user interaction).

Activation function: The user → system interaction (inputs) only takes place through widgets. Each user action on a widget may trigger one of the Cooperative Objects event handlers. The relation between user services and widgets is fully stated by the activation function that associates each event from the presentation part with the event handler to be triggered and the associated rendering method for representing the activation or the deactivation.

Rendering function: the system → user interaction (outputs) aims at presenting the state changes that occurs in the system to the user. The rendering function maintains the consistency between the internal state of the system and its external appearance by reflecting system states changes.

ICOs are used to provide a formal description of the dynamic behavior of an interactive application. An ICO specification fully describes the potential interactions that users may have with the application. The specification encompasses both the "input" aspects of the interaction (i.e. how user actions impact on the inner state of the application, and which actions are enabled at any given time) and its "output" aspects (i.e. when and how the application displays information relevant to the user).

An ICO specification is fully executable, which gives the possibility to prototype and test an application before it is fully implemented [24]. The specification can also be validated using analysis and proof tools developed within the Petri net community and extended in order to take into account the specificities of the Petri net dialect used in the ICO formal description technique. This formal specification technique has already been applied in the field of Air Traffic Control interactive applications [25], space command and control ground systems [32], or interactive military [5] or civil cockpits [2]. The example of civil aircraft is used in the next section to illustrate the specification of embedded systems.

To summarize, we provide here the symbols used for the ICO formalism and a screenshot of the tool.

- States are represented by the distribution of tokens into places 

- Actions triggered in an autonomous way by the system are represented as 

and called transitions
- Actions triggered by users are represented by half bordered transition 

ICOs are supported by the Petshop environment that makes it possible to edit the ICO models, execute them and thus present the user interface to the user and support analysis techniques such as invariants calculation.

3.2 Related Work on Formal Description Techniques and Usability

The use of formal specification techniques for improving usability evaluation has been explored by several authors.

Loer and Harrison [19] provide their view on how Formal Description Techniques (FDTs) can be used to support Usability Inspection Methods (UIM) such as the formative techniques discussed in the introduction of this paper. In accordance with our views, the authors argue that the costs of using FDTs are justified by the benefits gained. In their paper, the authors exploit the “OFAN” modeling technique [10], based on statecharts with the statemate toolkit for representing the system behavior. Each of the usability heuristics is formalized and tested however, the authors are limited to functional aspects of the system only.

In closer relation to the approach we present in this paper, Bernonville et al [7] combine the use of Petri nets with ergonomic criteria. In contrast to Loer & Harrison’s approach of assisting non-formal methods experts to benefit from FDTs, Beronville et al argue for the necessity of better communicating to computer scientists of ergonomic data.

Bernonville et al [7] use Petri nets to describe tasks that operators wish/should perform on a system as well as the procedure provided/supported by the software application. By using ergonomic criteria proposed by Bastien & Scapin [6], the method supports the analysis of detected problems. While Petri nets can be used to describe human tasks, dedicated notations and tool support such as ConcurTaskTrees (CTT) and ConcurTaskTree Environment (CTTe) [22] exist for the description of operator tasks, this would probably be more practical considering that "ErgoPNets" proposed in their paper currently has no tool support and is modeled using MS Visio.

Palanque & Bastide [31] have studied the impact of formal specification on the ergonomics of software interfaces. It is shown that the use of an object oriented approach with Petri nets ensures software and ergonomic quality. For example, by ensuring predictability of commands, absence of deadlocks and by offering contextual help and context-based guidance.

Harold Thimbley provides a worked example of a new evaluation method, called Interaction Walkthrough (IW), designed for evaluating safety critical and high quality user interfaces, on an interactive Graseby 3400 syringe pump and its user manual [38]. IW can be applied to a working system, whether a prototype or target system. A parallel system is developed from the interaction behavior of the system being evaluated.

4 Case Study

Before providing details of our approach, an outline of the case study on which we present our approach is provided here. The medical domain differs from the aviation domain (with which we are more familiar) in that it appears that there is more time to analyze a situation, for example, via contextual help. In a cockpit environment, it is

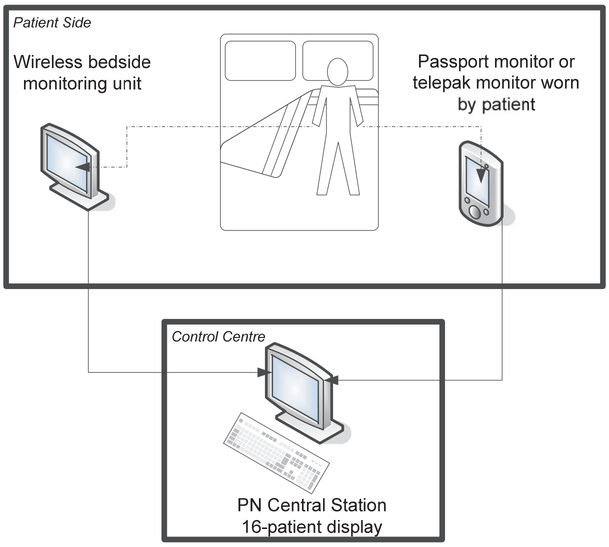

Fig. 4. Simplified layout of the PatientNet system

unlikely that the pilot or co-pilot will have time to access a contextual help service. The case study we have chosen is a telemetry patient monitoring system.

The PatientNet system, operating in a WMTS band, provides wireless communications of patient data from Passport monitors and/or telepak monitor sworn by patients, to central monitoring stations over the same network. Fig. 4 provides a simplified diagram of the layout of the PatientNet system.

Table 2. Summary of MAUDE adverse report search relating to PatientNet system

Patient Outcome Description	Number
Death	10
Unknown (was not specified in report)	8
Other	2
Required intervention	1
Life threatening	1
Total	**22**

Our interest in this system resulted from research on the US Food and Drug Administration (FDA) Manufacturer and User Facility Device Experience (MAUDE) database which has numerous adverse events reported (including fatalities) relating to this device/system. In our search, between the 20/03/2002 and the 08/02/2007, 22 reports were received relating to the PatientNet system including the central station and the passport. Of these 22, 10 resulted in patient death. Table 2 summarizes our findings in terms of patient outcome.

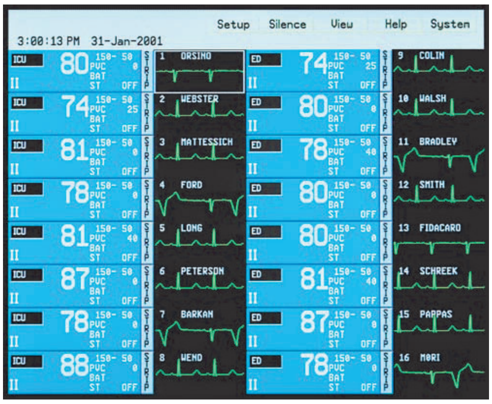

Fig. 5. PatientNet Central Station 16-patient display

The paper does not directly address incident and accident investigation, thus we will not continue our discussion on the adverse events that occurred during human-computer interaction with this system. We simply highlight the importance of such interactions and potential for abnormal human and technical behavior and the impact they may have on human life.

Furthermore, we will not describe the full system in detail; rather focus on the PatientNet central station for which we have screenshots of the application. The central station is a server and workstation where patient information from a variety of different vital sign monitors and ambulatory EGG (Electroglottograph) transceivers is integrated, analyzed and distributed to care providers. Real-time patient data is presented in a common user interface so caregivers have the information they need to respond to critical patient events immediately [39]. In addition to the patient's current status, the Central Station allows for retrospective viewing of patient information, which equips caregivers to make critical decisions and provide consistent, high-quality care to their patients across the enterprise (see Fig. 5 and Fig. 6 for Central Station screenshot).

Features of the central station include: Demographics, alarm limits, bedside view, event storage, ST analysis, full disclosure, trend storage, reports, care group assignments, physiological alarm volumes, print setup, choice of single or duel display.

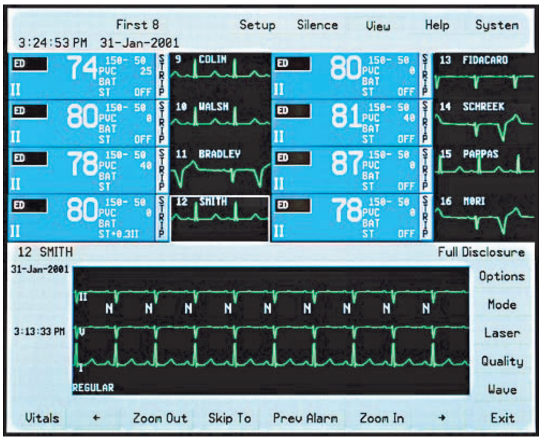

Fig. 6. PatientNet Central Station Full Disclosure

Typical selling points from the product brochure include statements such as "a simple touch of the screen brings up the information you need", "easy-to-read screen focuses attention on vital details" and "find answers to common questions with online Help menus".

5 The Approach

We demonstrate the applicability of this approach and how it could potentially predict and prevent some of the adverse events associated with a telemetry patient monitoring system made by a major medical technology manufacturer and currently used in a number of hospitals in North America.

Using the case study, we provide a simple example showing how formal description techniques and particularly the ICO notation and its Petshop tool, support the three points discussed in section 2 : usability evaluation, contextual help and incident and accident investigation. Each issue is a research domain within its own right and we are therefore limited to what we want to and will show on each point. The aim here is to give an overview of the types of support formal description techniques methods can provide to usability related methodologies/issues.

5.1 Supporting Usability Evaluation

As previously discussed, usability evaluation can be considered as formative or summative depending on the analysis technique used. Here we have decided to describe how the ICO FDT can be used to support a formative evaluation. The formative approach we use to illustrate this part of our approach is the application of ergonomic criteria and guidelines defined in [6], a type of usability inspection method. The selected criterion is "user explicit control" which has 5 guidelines. The selected guideline (number 4) is defined as "Allow users to interrupt or cancel a current transaction or process".

The PatientNet Central Station interface (see Fig. 5a, Fig. 5b) does not have a cancel or undo/redo option[2]. According to Bastien and Scapin's guideline, the user should be able to interrupt or cancel their current transaction. When the Central Station is in "full disclosure mode" (see Fig. 5b), an "exit" button is available to close the window and return to the 16-patient or 8-patient view mode. However, this "exit" button is not a cancel or an undo/redo function, it simply closes the window to change viewing mode.

Typically, if this problem was identified during a usability evaluation, the system would be redesigned, and a second round of user testing would be performed, to verify if all actions now have an interrupt option. Due to the highly interactive nature of such an application, it is impossible to fully verify if all available actions on the system can be interrupted, cancelled, or undone and redone.

The ICO formalism and its dedicated tool Petshop, can support the usability evaluation of such a guideline by providing a means to exhaustively analyze the system. Once the application's user interface has been modified (to provide a cancel or undo/redo button applicable to all transactions), the Petri net, using the ICO formalism is also modified to represent the same behavior. A link can then be made between the two components using the Activation (and/or rendering) function(s). Following this design modification stage, the benefits of FDTs can be exploited.

In addition to a second round of user testing, FDTs, such as ICOs provide a number of benefits for designers who wish to reason about the effects of proposed changes to a system. For example, a marking tree (an analysis technique based on mathematical foundations) can be used to identify the set of reachable states provided that some information is given about the initial status or 'marking' of the system. This technique helps to produce what can be thought of as a form of state transition diagram. Several tools are currently available to support this analysis (i.e. JARP Petri Net Analyzer version 1.1 developed by Sangoi Padilha (http://jarp.sourceforge.net/us/index.html) which has been developed as an auxiliary tool for the Petri net

[2] Access to the PatientNet system was not available for this study. We base our approach and examples on documented data. The point here is to illustrate the approach on a medical application.

analyzing tool ARP developed by Alberto Maziero (http://www.ppgia.pucpr.br/~maziero/diversos/petri/arp.html).

Thus in our example, marking graph can be used to trace all possible interactions with the system and **prove** that each action has a cancel or undo/redo option available.

The model provided in Fig. 7 represents the behavior of a small subsection of the PN central station system. The Petri net models the selection of modes (16-view, 8-view or full disclosure) and the availability of the cancel button. Assuming the initial state and default state is the 16-view mode, the cancel buttons are therefore available in 8-view mode and full disclosure mode. Furthermore, the user can switch between 8-view and full disclosure modes. We have modeled the system such that a "cancel" operation will take the system back into 16-view mode.

The places have been labeled p1 – p3 while the transitions have been labeled t1 – t6. Transitions that are fireable are shaded in dark grey. This from the initial state, 1 token in place p1, transitions t1 and t2 are the only two transitions that are fireable.

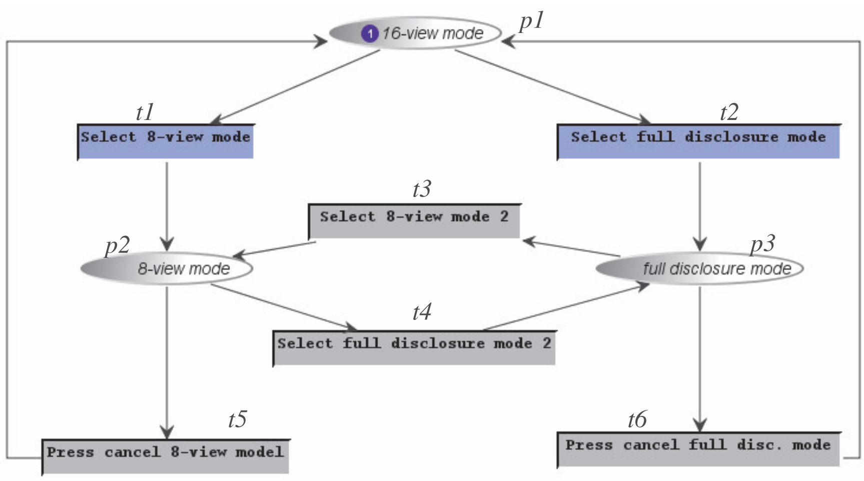

Fig. 7. ICO model proving a cancel option is always available

Since the model is simplistic, it is possible to analyze the model graphically. However, when modeling a complete system the use of systems' generated marking graphs is necessary to prove system properties. Here we provide the marking tree from the current marking of the model provided in Fig. 7. There are 3 possible states:

```
State M0: {p1}, State M1: {p2}, State M2: {p3}
The marking tree (represented in text format) is as follows:

M0: (t1:M1), (t2:M2) - From state M0, transition t1 changes the
current state to state M1, while transition t2 changes the current
state to M2

M1: (t4:M2), (t5:M0) - From state M1, transition t4 changes the
current state to state M2, while transition t5 changes the current
state to M0
```

```
M2: (t3:M1), (t6:M0) - From state M2, transition t3 changes the
current state to state M1, while transition t6 changes the current
state to M0
```

The analysis techniques described here provide exhaustive proofs of the availability of the cancel option in different system modes.

5.2 Supporting Contextual Help

The use of FDTs can provide accurate contextual help at any moment of system interaction. Since the model describes the concise behavior of the system, it is possible to analyze the exact state the system is in. As an example, we refer back to section 2.4, and discuss the case of an operator wishing to press a button to perform an action, though this button has no effect because it is currently "grayed out", inactive. This kind of interaction problem is encountered often with applications such as MS Word. Though after encountering the same problem numerous times, we begin to learn why the button is inactive and correct our actions accordingly. In the medical domain however, the first time this problem is encountered may result in life threatening scenarios. The operator should therefore be provided with contextual help that specifically states why the problem has occurred **and** how to resolve it. The FDT would provide a possible (or even an exhaustive list of all the possible) sequence of actions that would activate the required button, and a way of how to do it on the user interface but not whether it is wise to try to do it considering the current state of interaction between the user and the system and the current status of task execution.

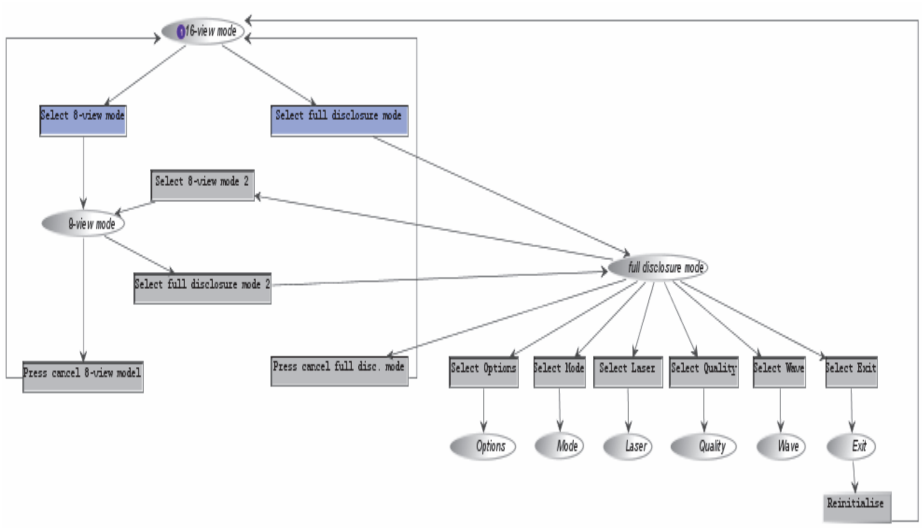

Fig. 8. ICO model with laser button inactive

Using ICOs, it is possible to analyze the Petri net and identify the current state (i.e. the distribution of tokens throughout the set of places) and understand why the button is inactive (i.e. which transition(s) must be fired in order for the place representing the button activation to contain its necessary token).

Fig. 8 and Fig. 9 illustrate a simple Petri net, using the ICO notation, in order to demonstrate this part of the approach. We take the same model illustrated in the previous section, this time adding the options available when in full disclosure mode. These include options, mode, laser, quality, wave and exit. The exit transition takes the system back into the 16-mode view. Fig. 9 indicates that the "laser" button is currently not available, because its associated transition is not fireable.

The only way that the "select laser" transition can become fireable is for place "full disclosure mode" to contain a token. If the user was searching for the laser button while operating the PN Central station and referred to a help file, the Petri net model could provide detailed help on why the button is not available and how to make it available. Fig. 9 therefore shows the same model, this time in the state allowing the "select laser" transition to be fireable.

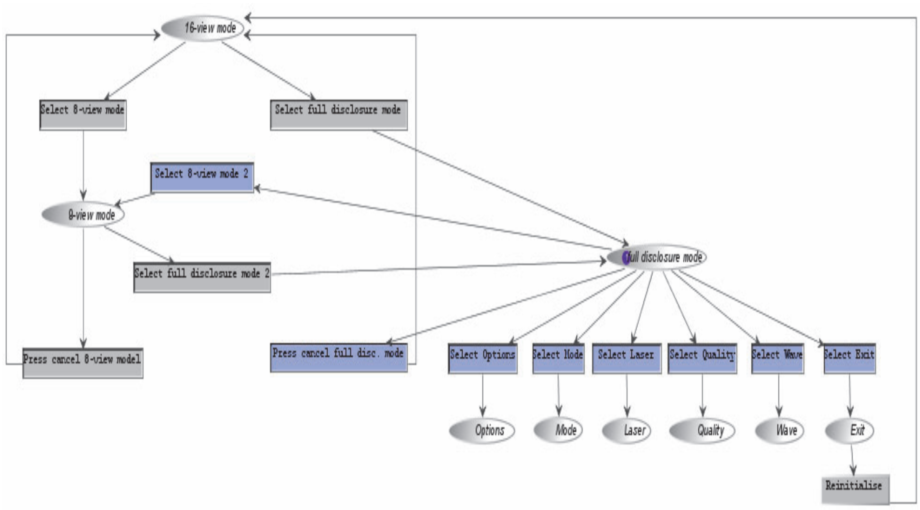

Fig. 9. ICO model with active laser button

5.3 Supporting Incident and Accident Investigation

In order to support the investigation of an incident or accident, such as the reports on the US Food and Drug Administration (FDA) Manufacturer and User Facility Device Experience (MAUDE) database , we advocate the use of system modelling to prove that a given incident or accident cannot be triggered in the new system design. We are not claiming that the mishap will not recur in the real system, but in the model describing the behavior of the system. This model is an abstraction of the real system assuming that system engineers will use that model for designing the improved system.

To be more concrete, we have devised an approach that allows a system to be formally modeled and then formally analyzed to prove that the sequence of events described in the accident report is not able to be triggered again in that model.

The aim is to use formal description techniques to model the complete behavior of the system and identify hazardous states that we would wish to avoid. These states can be avoided by eliminating the sequence of state changes leading to that state.

One of our aims for including data from incident and accident investigations in safety-critical interactive systems design is to ensure that the same incident or accident analyzed will not occur again in the re-modeled system. That is, that an accident place within the network will be blocked from containing a token. Due to space constraints we do not illustrate this part of the approach in this paper, though the interested reader can refer to [3] and [4].

6 Conclusion

This paper has addressed issues of usability for modern medical applications. We have argued that empirical usability evaluations, such as formative and summative evaluations are not sufficient when dealing with complex safety-critical interactive systems that have an extremely large number of system states. We advocate the use of formal methods, particularly those based on Petri nets, such as the Interactive Cooperative Objects (ICOs) formalism, a formal notation dedicated to the specification of interactive systems that can be exploited to provide mathematically grounded analyses, such as marking graphs, to argue and prove usability properties. By producing high-fidelity formally specified prototypes, it is possible to verify whether the new design satisfactorily prevents such ergonomic defects. The paper has given three examples of ways in which Formal Description Techniques (FDTs) can support usability. However, each of these examples (formative evaluation, contextual help and supporting incident and accident investigation) is its own research area, and here we give a taster of each.

Acknowledgments. This work was supported by the EU funded Network of Excellence ResIST http://www.resist-noe.eu.

References

1. Adolf, J.A., Holden, K.L.: Touchscreen usability in microgravity. In: Tauber, M.J. (ed.) CHI 1996. Conference Companion on Human Factors in Computing Systems: Common Ground, ACM Press, New York (1996)
2. Barboni, E., Navarre, D., Palanque, P., Basnyat, S.: Exploitation of Formal Specification Techniques for ARINC 661 Interactive Cockpit Applications. In: HCI Aero 2006. Proceedings of HCI aero conference, Seatle, USA (September 2006)
3. Basnyat, S., Chozos, N., Palanque, P.: Multidisciplinary perspective on accident investigation. Reliability Engineering & System Safety 91(12), 1502–1520 (2006)
4. Basnyat, S., Chozos, N., Johnson, C., Palanque, P.: Redesigning an Interactive Safety-Critical System to Prevent an Accident from Reoccurring. In: 24th European Annual Conference on Human Decision Making and Manual Control (EAM) Organized by the Institute of Communication and Computer Systems, Athens, Greece (October 17-19, 2005)
5. Bastide, R., Palanque, P., Sy, O., Le, D.-H., Navarre, D.: Petri Net Based Behavioural Specification of CORBA Systems. In: ATPN 1999. International Conference on Application and Theory of Petri nets. LNCS, Springer, Heidelberg (1999)

6. Bastien, C., Scapin, D.: International Journal of Human-Computer Interaction. 7(2), 105–121 (1995)
7. Bernonville, S., Leroy, N., Kolski, C., Beuscart-Zéphir, M.: Explicit combination between Petri Nets and ergonomic criteria: basic principles of the ErgoPNets method. In: EAM 2006. European Annual Conf. on Human Decision-Making and Manual Control, Universitaires de Valenciennes (2006) ISBN 2-905725-87-7
8. Bernhaupt, R., Navarre, D., Palanque, P., Winckler, M.: Model-Based Evaluation: A New Way to Support Usability Evaluation of Multimodal Interactive Applications. In: Law, E., Hvannberg, E., Cockton, G. (eds.) Maturing Usability: Quality in Software, Interaction and Quality, Springer, London (2007)(accepted for publication)(to appear)
9. Bredereke, J., Lankenau, A.: A Rigorous View of Mode Confusion. In: Anderson, S., Bologna, S., Felici, M. (eds.) SAFECOMP 2002. LNCS, vol. 2434, pp. 19–31. Springer, Heidelberg (2002)
10. Coutaz, J., Bérard, F., Carraux, E., Crowley, J.L.: Early Experience with the Mediaspace CoMedi. In: Proceedings of the IFIP Tc2/Tc13 Wg2.7/Wg13.4 Seventh Working Conf. on Engineering For Human-Computer interaction, pp. 57–72. Kluwer Academic Publishers, Dordrecht (1999)
11. Fourney, D., Carter, J.: Proceedings of GOTHI-05 Guidelines On Tactile and Haptic Interactions. In: Fourney, D., Carter, J. (eds.) USERLab, Univ. of Saskatchewan (2005), http://userlab.usask.ca/GOTHI/GOTHI-05 Proceedings.html
12. Furnas, G.W.: Generalized Fisheye Views. In: Proc of ACM CHI 1986, pp. 16–23. ACM, New York (1986)
13. Holzinger, A.: Usability Engineering for Software Developers. Communications of the ACM 48(1), 71–74 (2005)
14. Holzinger, A., Errath, M.: Designing Web-Applications for Mobile Computers: Experiences with Applications to Medicine. In: Stary, C., Stephanidis, C. (eds.) User-Centered Interaction Paradigms for Universal Access in the Information Society. LNCS, vol. 3196, pp. 262–267. Springer, Heidelberg (2004)
15. Holzinger, A.: Finger Instead of Mouse: Touch Screens as a means of enhancing Universal Access. In: Carbonell, N., Stephanidis, C. (eds.) Universal Access. Theoretical Perspectives, Practice, and Experience. LNCS, vol. 2615, pp. 387–397. Springer, Heidelberg (2003)
16. Holzinger, A., Sammer, P., Hofmann-Wellenhof, R.: Mobile Computing in Medicine: Designing Mobile Questionnaires for Elderly and Partially Sighted People. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 732–739. Springer, Heidelberg (2006)
17. Kohn, L., Corrigan, J., Donaldson, M.: To Err Is Human: Building a Safer Health System. In: Institute of Medicine.Committee on Quality of Health Care in America, National Academy Press, Washington DC (1999)
18. Lamping, J., Rao, R.: Laying out and visualizing large trees using a hyperbolic space. In: ACM Symp User Interface Software and Technology, pp. 13–14. ACM Press, New York (1994)
19. Loer, K., Harrison, M.: Formal interactive systems analysis and usability inspection methods: Two incompatible worlds? In: Palanque, P., Paternó, F. (eds.) DSV-IS 2000. LNCS, vol. 1946, pp. 169–190. Springer, Heidelberg (2001)
20. Mackinlay, J.D., Robertson, G.G., Card, S.K.: Perspective Wall: Detail and Context Smoothly Integrated. In: Proceedings of SIGCHI 1991, pp. 173–179 (1991)

21. Memmel, T., Reiterer, H., Holzinger, A.: Agile Methods and Visual Specification in Software Development: a chance to ensure Universal Access. In: Coping with Diversity in Universal Access, Research and Development Methods in Universal Access. LNCS, vol. 4554, Springer, Heidelberg (2007)
22. Mori, G., Paternò, F., Santoro, C.: CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. IEEE Transactions on Software Engineering , 797–813 (August 2002)
23. Navarre, D.: Contribution à l'ingénierie en Interaction Homme Machine - Une technique de description formelle et un environnement pour une modélisation et une exploitation synergiques des tâches et du système. PhD Thesis. Univ. Toulouse I (July 2001)
24. Navarre, D., Palanque, P., Bastide, R., Sy, O.: Structuring Interactive Systems Specifications for Executability and Prototypability. In: Palanque, P., Paternó, F. (eds.) DSV-IS 2000. LNCS, vol. 1946, Springer, Heidelberg (2001)
25. Navarre, D., Palanque, P., Bastide, R.: Reconciling Safety and Usability Concerns through Formal Specification-based Development Process. In: HCI-Aero 2002, MIT Press, USA (2002)
26. NHS Expert Group on Learning from Adverse Events in the NHS. An organisation with a memory. Technical report, National Health Service, London, United Kingdom (2000), http://www.doh.gov.uk/orgmemreport/index.htm
27. Nielsen, J., Molich, R.: Heuristic evaluation of user interfaces. In: Proceedings of CHI 1990, pp. 249–256. ACM, New York (1990)
28. Nielsen, J.: Heuristic evaluation. In: Nielsen, J., Mack, R.L. (eds.) Usability Inspection Methods, John Wiley & Sons, New York (1994)
29. Nielsen, J. (2005) http://www.useit.com/papers/heuristic/ last accessed 21/06/2007
30. Norman, D.A., Draper, S.W. (eds.): User-Centred System Design: New Perspectives on Human-Computer Interaction. Lawrence Earlbaum Associates, Hillsdale (1986)
31. Palanque, P., Bastide, R.: Formal specification of HCI for increasing software's ergonomics. In: ERGONOMICS 1994, Warwick, England, 19-22 April 1994 (1994)
32. Palanque, P., Bernhaupt, R., Navarre, D., Ould, M., Winckler, M.: Supporting Usability Evaluation of Multimodal Man-Machine Interfaces for Space Ground Segment Applications Using Petri net Based Formal Specification. In: Ninth International Conference on Space Operations, Rome, Italy (June 18-22, 2006)
33. Palanque, P., Farenc, C., Bastide, R.: Embedding Ergonomic Rules as Generic Requirements in a Formal Development Process of Interactive Software. In: proceedings of IFIP TC 13 Interact 99 conference, Edinburg, Scotland, 1-4 September 1999 (1999)
34. Palanque, P., Bastide, R., Dourte, L.: Contextual Help for Free with Formal Dialogue Design. In: Proc. of HCI International 93. 5th Int. Conf. on Human-Computer Interaction joint with 9th Symp. on Human Interface (Japan), North Holland (1993)
35. Pierotti, D.: Heuristic Evaluation - A System Checklist, Xerox Corporation (1995), Available onlineat http://www.stcsig.org/usability/topics/articles/he-checklist.html
36. Scapin, D. L.: Guide ergonomique de conception des interfaces homme-machine (Rapport de Recherche No. 77). INRIA - Rocquencourt – France (1986)
37. Spence, R., Apperley, M.: Data Base Navigation: An Office Environment for the Professional Behaviour and Information Technology 1(1), 43–54 (1982)
38. Thimbleby, H.: Interaction walkthrough - a method for evaluating interactive systems. In: The XIII International Workshop on Design, Specification and Verification of Interactive Systems, July 26-8, 2006 Trinity College, Dublin(2006)
39. http://www.gehealthcare.com/usen/patient_mon_sys/wireless_and_telemetry/products/telemetry_sys/products/patientnet_centralstation.html