Adapting Interface Representations for Mobile Support in Interactive Safety Critical Contexts

Fabio Paternò, Carmen Santoro, David Touzet

ISTI-CNR Pisa, Italy.
http://giove.isti.cnr.it

Abstract: Mobile technology is penetrating many areas of human life. However, little attention has been paid so far to its use and impact in interactive safety critical contexts. We present a method that aims to provide designers with a better understanding of the introduction of mobile devices in such contexts and help them to identify and derive interfaces that support users in their activities. The method is a novel combination of a systematic analyses of potential deviations in task performance and information representations based on distributed cognition. The results of the conceptual design can drive model-based transformations able to identify and implement suitable interface representations. The originality of the contribution is in combining the results of a distributed task performance analysis with a transformation-based approach for generating user interfaces able to take into account such results.

Keywords: user interface, safety-critical systems, mobile devices

Introduction

In recent years there has been an increasing availability and use of a wide range of interactive devices, in particular mobile devices. This type of technology is penetrating many areas of human life. In interactive safety critical systems the introduction of new technology is often slow because people need to carefully understand their implications in terms of potential hazards. Only recently these issues have started to be addressed (Buisson and Yannick, 2001).

In this paper we present a novel method that aims to help designers in understanding such implications and finding the suitable representations that can be provided through mobile devices in order to improve usability while preserving safety.

Our method (analysis of distributed task performance) is based on the integration of systematic analysis of deviations and analysis of information representation based on the Distributed Cognition approach (Hitchins, 1995), (Hollan, Hutchins, Kirsh, 2000) (Fields *et al.*, 1998). In deviations analysis there is a systematic analysis of potential effects in case of *deviations* from the task plan. In order to help with such analysis, a number of deviation types (indicated by *guidewords*) have been identified. The approach is supported by task models, which are suitable to providing an overall view of the possible activities but may not be able to capture all the possible contextual aspects. This information can be provided through the support of Distributed Cognition analysis. It focuses on how knowledge is distributed across individuals, tools and artefacts in the considered context/environment. One basic point is that the breakdown in task performance is the consequence of inadequate access to the distributed representation of information resources supporting task performance. One limitation of this approach is the difficulty of translating its results into specific design criteria. The integration with the analysis of tasks and their performance can create the basis for addressing this issue. Once a better understanding of the tasks to accomplish and their information needs has been achieved, the third main element of our method comes into play: we apply a model-based approach (Paternò, 1999) (Wilson *et al.*, 1993) to the design of interfaces for a variety of devices, including mobile devices, taking into account the tasks to accomplish and the information regarding the context of use. In this process the idea is that the final representations provided should be suitable for the activities to support but can radically differ depending on the interaction resources available on the device at hand.

In the paper we describe and discuss the proposed method and show its application in a real safety-critical context, an Air Traffic Control Centre, for which we analyse the possible use of mobile interactive devices for supporting a specific role and some activities.

The Case Study

In order to show a potential application of the approach described and understand its feasibility, we considered a real application domain, the Air Traffic Control, and we extracted a case study in the working environment of Rome-Ciampino control centre in Italy. In this control room, there is a number of en-route and approach working positions in charge of controlling respectively cruising flights and airplanes taking off/landing to the nearby major Fiumicino airport. In addition, there are other stakeholders (a chief controller, a technician supervisor, a flow controller), together with three or more supervisors having the responsibilities for making decisions about closing/opening sectors (usually in a vertical manner), depending on data about the estimated traffic load and airport capacity.

Air Traffic Management is based on the concept of airspace division into a number of sectors. The number of flights that are planned to cross a specific sector is called traffic demand. The maximum number that may be in a certain sector simultaneously (namely the number of flights that the sector itself is able to handle for each hour), is called traffic capacity and can be calculated using a number of parameters (types of flights, air complexity, etc.). If the number of flights that intend to cross a specific sector (traffic demand) is higher than the number of flights that the sector itself is able to handle (traffic capacity), a controller (flow controller) requires the emission of a flow or a so-called regulation. When a flight is subject to a regulation, a time slot in which a flight should take off is assigned, so as to distribute the traffic within a broader interval of time (while still maintaining the requests within the capacity of the sector).

More specifically, when the controllers in charge of the flow position recognise a situation in which the traffic demand exceeds the sector capacity, they coordinate with the supervisors and chief controller, and decide the actions to be taken (e.g.: some flights might be redirected through a different path with the same length, or a regulation might be triggered). In the current system, controllers use graphs visualised on paper-based documents such as that displayed in Figure 1, to analyse the variation in the traffic demand during (a part of) the day. As can be seen from the picture, the threshold line for the capacity of the concerned sector is 40 flights, which means that the sector is supposed to manage a maximum number of 40 flights per hour. However, there are some peak hours during the day in which the expected traffic might exceed such a limit: indeed in the time interval 11:00-15:00 it might occur more than once that the traffic demand is higher than the capacity; so appropriate action (e.g.: a regulation) should be triggered by the controllers.

In a situation of critical meteorological conditions, there is a controller in the centre, who *manages information about the upcoming traffic* in order to trigger de-combining of two sectors. Indeed s/he has the responsibility of making decision about closing/opening sectors (usually dynamically divided in a vertical manner), depending on data about the estimated traffic size and the airport capacity, and also personnel resources available on site.

The ATC supervisor can be regarded as the only role without any “dedicated” position within the control room. The need to have permanent access to real time traffic information may imply a high level of mobility in the control room.

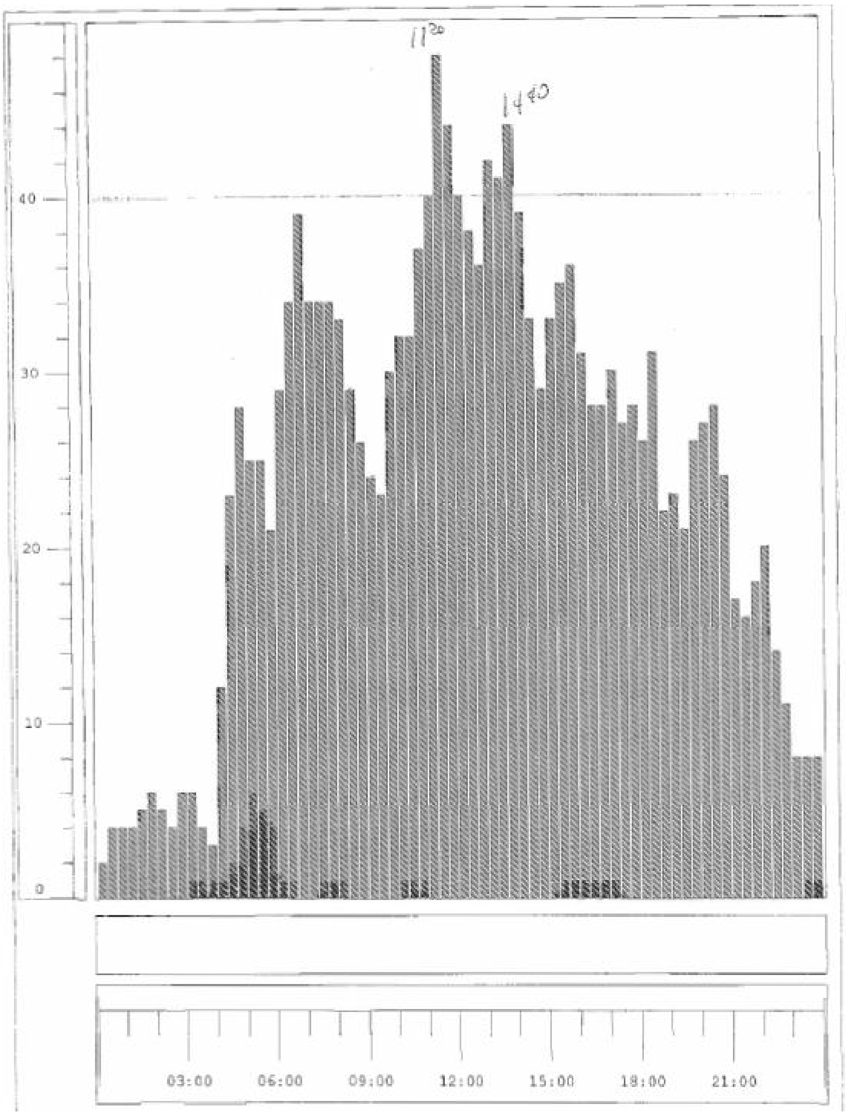

Figure 1 - A Working Tool for the Flow Controller in the Current Environment: Foreseen Traffic Displayed with Paper Bar Charts

In order to de-combine two air space sectors, the supervisor has to identify overloaded air traffic sectors, and the level of criticality of the upcoming air traffic; at the same time, he has to evaluate the on-site workload allocation of the controllers, and also to identify the personnel available to assume control of a new sector. The complex information supporting the supervisor's task is available from several sources distributed in the task space: flight information system, air traffic monitoring system, radar, flight progress strips, meteorological information, etc.

In this scenario the ATC controllers have to decide whether to open a new sector or not by checking the critical threshold of the upcoming air traffic level. In order to perform their activities, they use an integrated set of tools (computer displays, telephones, paper-based documentation). In addition, they need real time access to information about the current and estimated up-coming air traffic levels, as the decision to open a new sector will be further based on the information manipulated by this task.

The Methodological Approach

The basic idea is to evaluate a current design in order to analyse and identify possible areas for improvement, in terms of some criteria which should also depend on the specific application considered. For instance, if we consider a safety-critical application, a possible analysis should perform risk assessment in the current design, namely understanding and quantifying the level of risk involved in the considered design in order to evaluate if it is acceptable/tolerable or new actions should be provided to manage hazardous situations that might occur. As soon as such areas of improvements have been identified, the next step is to identify and specify a new arrangement/distribution of activities, roles, artefacts and devices in order to compensate for the identified shortcomings. Indeed, such specification should describe how the activities are supposed to be carried out in the new system, also specifying how they evolve over time, the context in which they are supposed to be performed, the roles that are expected to act, and the artefacts and/or devices available in the new setting.

In the analysis of potential improvements it is also possible to consider the impact of new technology, such as mobile devices. Once a new model of how activities should be organized has been identified in such a way as to address the issues of the previous solution, it can be captured in a specification that can be used to derive a new user interface able to take into account the new requirements. This can be obtained with the support of model-based design and development tools able to consider the logical descriptions of the activities to perform and suggest and generate user interfaces suitable for their support, even considering the possible interaction platforms.

Such process can then be iterated because, once a prototype of the new user interface is produced, it might in turn be subject to evaluation (according to the analysis of distributed task performance) and, as a result of such analysis, further changes to the specification might be included, so as to restart the process.

Analysis of Deviations

In this section we provide a more detailed description of the method proposed. One starting point for this research has been the deviation analysis (Paternò and Santoro, 2002). In executing the analysis, several aspects need to be carefully identified: role, task, representations, and deviations.

1. Analysis of the task and related properties;
2. Analysis of the representations associated with the task
3. Analysing deviations from the task plan

The results of such analysis may be stored in a table with the following information:

- *Task*: the activity currently analysed, together with some properties relevant to our analysis;
- *Representation distribution*: the resources supporting task performance and their distribution;
- *Guideword*: the type of interaction failure considered;
- *Explanation*: how an interaction failure has been interpreted for that task and the given deviation;
- *Causes*: the potential causes for the interaction failure considered and which configuration of resources might have generated the problem;
- *Consequences*: the possible effects of the interaction failure in the system;
- *Recommendation*: suggestions for an alternative (if any) distribution of resources able to better cope with the considered interaction failure.

The deviations analysis can be applied to all possible activities, even if the detailed consideration of causes, consequences and recommendations makes particularly sense in the case of safety critical systems. By way of example, we can consider a simple activity such as printing and a deviation type such as None. This means that the printing has not effect and we have to understand why this happens. This could be for several reasons: no input information (the user did not correctly select the file to print) or the activity is not performed (the printer is broken) or the activity is performed but does not generate the result (the printer is out of paper).

If potential safety-critical issues have been identified then our analysis aims at identifying better representations (or distributions of them) that could be more suitable for carrying out the considered tasks and prevent occurrences which can have safety-critical effects. The evaluation has to consider if a different allocation of resources may be envisaged, which implies different representations of information and could involve considering different devices, that may result in a significant improvement for the overall system's safety and usability.

Analysis of Information Representation

In this part of the method the focus is on the analysis of the information representations in order to understand whether they are optimal for the distributed task performance. To this end, a number of important attributes have been identified:

Externalisation: to what extent the representation is explicitly provided in the real world or is based on implicit, internal representations..

Accessibility: whether it can be problematic for the user to access the information because it is difficult to see or to hear it or for any other reason.

Access modality: the type of access that a user has to a representation. Different types of access (sequential/concurrent) could be exploited for externally available representations.

Mobility: whether the access to the information requires the user to move or the user can move while accessing information.

Sharing: refers to the extent the perception of a representation is: i) Local to individuals; ii) Shared (e.g. by the members of a team); iii) Globally available to all. This property might be connected with the type of supporting platform (for example if controllers annotate a strip on their PDAs, this information will be available locally to them).

Persistence: whether transient or permanent access to information is allowed.

Flexibility of modifying the representation, ability to flexibly update and modify the representation, for example allowing a person to annotate an external representation (i.e. strips).

Operations and actions supported such as:

Comparability: with other objects / representations available in the user's context;

Combinability: allowing users to combine information from different sources;

Ease of production: allowing reconfiguring and multiple views of information;

If this type of analysis is applied to our case study we can notice that several information objects supporting task performance may be identified: normal and critical threshold of upcoming air traffic level, additional parameters such as estimated numbers of aircraft, together with time intervals, planned trajectories, etc. In particular, such thresholds of upcoming air traffic level are visualised on paper in the form of a bar chart, such as that visualised in Figure 1. As for the properties associated to the different resources, various properties have been identified:

Externalisation: they are available in both graphical and numerical representation forms.

Accessibility and Access Modality: the representation forms and media (in this case, large computer screens) allow users concurrent, easy access to a variety of information. If accessing the same information with a PDA, it is expected that its physical constraints (i.e. screen size) will make sequential the access to information, therefore increasing the time and effort needed to visualise the same items. On the other hand, a PDA would allow permanent access to the required information, even if user changes his position across the control room.

Mobility: The user is expected to move about the control room to access the needed information.

Sharing: similarly, using a small screen device is likely to change the observability of information, from being easily *shared* with other members of the team, to *locally* available to the user of the device.

Persistence: critical information (e.g.: threshold of the upcoming air traffic level) is graphically represented, thereby allowing non-transient access.

Flexibility of cognitive tracing and interactivity. In the considered case, the parameters of interest are changing *autonomously* according to the real-time situation of the air traffic flow. Controllers have no or minimal permission to effect changes, to annotate or update an external electronic representation. A standard working position is equipped with no input device (i.e., a keyboard), as only direct manipulation of the objects already available on the screen is allowed.

Operations and actions supported

Comparability. the graphical representation of information employed (i.e., clustered columns) provides users with the possibility to directly *compare* various values of the monitored variables (i.e., by rapidly perceiving differences between the height of two columns).

Combinability - possibility to combine and reconfigure or re-represent the information of interest: well supported in the current ATC work settings. For instance, the information contained in an electronic flight strip can be displayed in two different formats; the values of the upcoming traffic may be represented graphically as well as numerically, etc. For the hypothetical situation of using a PDA, the question is how to effectively display the relevant information in the perimeter of a very small screen space, while maintaining a high level of interactivity. For instance, a solution would be to reduce the amount of graphics, rely mainly on the numerical representation of information, and using additional codes (e.g. sounds) in order to facilitate user's rapid discrimination of critical information.

Analysing Deviations from the Task Plan

It is possible to apply the deviation analysis to the case study to identify potential safety-critical issues. For instance, if we consider the possibility that the representation of interest is not available, there are some possible causes that might be identified. For instance, *the information is not visible* and the possible causes are that there are some difficulties in perceiving the relevant information due to some usability issues (the object represented is too small, ambiguous shape, wrong choice of colour, etc.; but also supervisor' s distraction / interruption by other activities, etc.). In other cases, *the representation might not be persistent*, due to rapid change of information values, which does not give the user the necessary time to internalise the perceived information and to integrate it with the other information supporting his decision making.

As for possible consequences, if there is no information available in the task performance space (not visible, not persistent) then various types of task failure can occur (e.g. stop task performance, delay, etc.). Then, as a result of this analysis, a possible r*ecommendation* might be to rely on multiple ways of representing the same information (e.g., visual and auditory): access to concurrent representation of the same information could be especially important for users 'on the move', who allocate their attention to several competing tasks. Design should facilitate a rapid perception of the relevant information, and support an accurate interpretation of its significance (e.g., estimation of the air traffic flow - under/ over a critical level, or its approximate value). For instance, discrimination of the critical information could be facilitated by suitable use of colour, use of multimedia facilities such as animation, blinking images, the use of sound, etc.

Redesigning the Work Model

As the first step of the analysis has highlighted the need for the controller to have information available while moving round in the control room, then, a new specification of the activities that should be carried out is to be described, in order to solve the problem that has been identified. In the new, envisioned system, the activities should be carried out so as to allow controllers constant access to the information that is needed to perform their activities. The new specification should identify the new context in which the activities are carried out, and the new arrangements of resources/devices.

More specifically, the envisioned system calls for providing the controller with a mobile device to display the critical information. The controller needs to access such information in real time, so it should be always available, and, to this end, the possibility that such information should be visualised on a PDA might be envisaged in the new system. Then, in this case, due to the limited capabilities of the handheld device, only a selected subset of the data normally displayed by the current system tools should be visualised on such devices. In addition, due to the new context of information displayed on the PDA (the user is supposed to be mobile), specific presentation techniques able to cope with the rather noisy environment of the control room should be foreseen and the eventuality that the controller not watch the device constantly be adequately controlled for.

In addition, special attention should be paid on how to render some specific types of representation (like those used in the ATC case study considered, eg bar charts and line charts) in order to understand the best way to render such data on the small screens available on handheld devices such as PDAs.

In our new approach, once the activities have been specified, together with information about the new arrangements of tasks and resources, such specification represents the input from which a new design of the user interface can be derived. This information should be specified so as to address the characteristics of the problem in a top down manner, from the abstract level and then refining to more concrete levels. Indeed, the process allowing to derive the user interface from a high level description of how the activities are supposed to be carried out in the new environment includes three basic steps, that will be described in more details in the following subsection.

Linking Design and Development

The method for generating the user interface is based on three main transformations. Such transformations have been implemented in a tool, TERESA, which is a semi-automatic environment supporting a number of transformations useful for designers to build logical descriptions and exploit the information that they contain to consequently generate the user interface for various types of platform.

The main abstraction levels considered are: the task model, where the logical activities to support are identified and the abstract user interface, a logical description of the user interface. They are used to obtain a user interface implementation able to support effectively the tasks identified. Then, there is a concrete level, which is useful to link the abstractions and the implementation. The main transformations considered are:

- *From Task Model-related Information to the Abstract User Interface*. The task model specification, along with information regarding groups of tasks that are enabled over the same period of time, are the input for the transformation generating the associated abstract user interface, which will be described in terms of both its static structure (the presentation part) and dynamic behaviour (the dialogue part). The structure of the presentation is defined by logical interaction objects (interactors) characterized in terms of the basic tasks that they support, and their composition operators. Such operators aim to structure logically the presentation according to the communication effects desired. They are grouping, which indicates a set of interface elements logically connected to each other; relation, highlighting a one-to-many relation among some elements, one element has some effects on a set of elements; ordering, which indicates that some kind of ordering among a set of elements can be highlighted, and hierarchy, which is used when different levels of importance can be defined among a set of elements.

- *From the Abstract User Interface to the Concrete User Interface*. This transformation starts with the abstract user interface; it is possible to move into the related concrete user interface for the selected specific interaction platform. The difference between these two levels is that the abstract description is modality and platform independent whereas the concrete description is platform dependent. Thus, for example at the abstract level the designer can specify that there is a need for a selection object whereas at the concrete level the specific interaction technique is indicated (for example, a radio-button or a list or a vocal selection). It is worth pointing out that the concrete level is still a logical description and is independent from the specific implementation language or device which is going to be used. Indeed, the platform is a characterization of a group of devices that share similar interaction resources (such as the desktop, the vocal device, the PDA and so on). A number of parameters related to the customization of the concrete user interface are made available to the designer in order to obtain the concrete interface, with different levels of intervention required from the designer, ranging from completely automatic solutions to other cases in which the designer might modify all the possible details in the design process. The tool can provide suggestions according to predefined design criteria, but developers can modify them. In addition, depending on the type of platform considered there are different ways to implement design choices at the user interface level.

- *From the Concrete User Interface to the Final User Interface*. This last step generates the interface code according to the type of platform selected from the concrete user interface. Before generating the final code, it is possible to specify within the tool the value of additional parameters allowing the designers to still diversify between the various devices belonging to the same platform. For instance, if we select to generate the final code for a mobile platform, depending on the features of the current device, a different final user interface will be produced by the tool e.g. in XHTML, XHTML Mobile Profile, SVG and VoiceXML.

Rendering Interactive Graphical Representations

In order to better support the results of a distributed task performance analysis, a new version of the TERESA environment (Mori, Paternò, Santoro, 2004) has been designed. The main new features consist in the possibility of generating interactive graphical representations (implemented in SVG) that adapt to the feature of the devices considered, including mobile devices, following different strategies. This means that a novel transformation from the abstract user interface level and the implementation has been designed for this purpose.

The abstract user interface is generated from the task model. This transformation is based on the information associated with each user task and the temporal and semantic relation among them. Thus, abstract structured objects compose elementary objects according to some logical information. For instance, at the abstract level structured tables are considered as structured objects composed of lists of basic elements (numerical, textual elements…), so they are modelled as a set of ordered lists characterized by the type of data that they contain, and associated each other by means of some abstract relations that model the semantic link existing between them. The various ways in which such relations occur at the abstract level (we call them abstract operators) are translated, at the concrete level, into appropriate techniques able to convey the related meaning also at this level. For example, concrete chart encoding like pie chart, bar charts, etc are all examples of concretely translating abstract combinations of list of ordered elements (it is an example of application of what we called the *objectOrdering* abstract operator).

Concrete user interfaces are basically computed based on the structure of abstract user interfaces together with additional information useful for selecting the most appropriate concrete rendering technique, depending on the platform that will be used to render the user interface. Indeed, moving from the abstract to the concrete level first requires the designer to select a target device type among those supported by the environment.

These concrete models associate each structured object with a specific concrete chart encoding (bar chart, line chart…) of data. These techniques have to be consistent with the previously declared abstract types. Among the relevant representations that are possible (for example bar charts, line charts, and scatter plots), if some may not be rendered correctly for any reason (such as insufficient colour or interactivity support) on the target platform then they are not considered. According to the selected chart type, designers should be able to specify a number of additional information in order to customise the presentation, so as to fine-tune better the layout of the chart that will be generated. For instance, with charts for desktop environments, the designer can decide whether numerical values have to be displayed on the chart in addition of their graphical encoding.

In our case, and with the specificity of the case study considered, the constraints inherent to small screen enabled devices are addressed by the generation of interactive structured graphics embedding dynamic exploration functionalities. By exploiting some information visualization techniques, the interactive exploration facilities aim to enable users to efficiently access information encoded by structured graphics on small displays.

As interactive exploration facilities depend on the abstract structure of the considered object, a technique that at the abstract level allows the designer to specify the relative importance of basic elements (we called it *objectHierarchy* abstract operator) will be associated, at the concrete level, to an interaction technique able to highlight differently the information depending on the available capabilities of the device (for instance through the use of techniques such as semantic zooming). In addition, an *objectHierarchy* operator at the abstract level can be also rendered through the use of fisheye view exploration mechanism, able to highlight the most important information to be presented depending on the Degree Of Interest dimension.

Analysis of the Resulting Representations in the Example

The application of our approach to the case study can find a more efficient manner of showing data needed to the ATC controller in a PDA-enabled new system, as a consequence of ensuring a greater level of safety in situations such as that highlighted by the deviation-based analysis previously performed, in which the controller might be temporarily unaware (because, e.g. is distant) of some critical information currently visualised on some tools.

In the new, envisaged system, the mobile device enables the controllers to move around the control centre bringing with them the device so as to get a full, continuous awareness of the expected situation. In addition, thanks to the fisheye view-equipped graphs the controllers have on their PDAs, it is possible for them to more properly focus on the current area of interest, which are the intervals of time when the threshold limit are likely to be overcome. Moreover, there is the possibility for the controller to have additional information on specific time intervals, by tapping-and-holding the pen stylus on some specific bars, so as to have visualised more precise information on the concerned values. For instance, in Figure 2, the controllers have currently focused their interest on the period of time between 12:40 and 13, and a tooltip is displayed in order to more precisely show that the number of flights that are expected for that period of time is 42, which is beyond the supposed capacity of the sector.

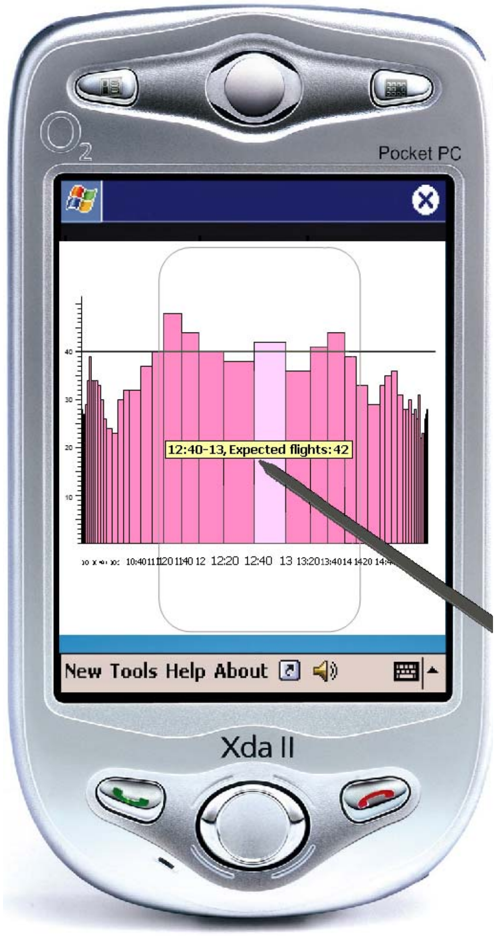

Figure 2 - The Fisheye Viewed Bar-Chart with Flight Information for PDA.

The resulting presentation is able to address some issues highlighted by the deviation-based analysis performed in the first stages of our approach. First of all, the introduction of a mobile device in the new environment allows the controllers to get anytime the necessary information for performing their work, so compensating to the lack highlighted when the controller is far from the stationary workstations and then the concerned information is out of reach. In addition, the use of effective information visualisation techniques on handheld devices has made the rendering of such critical information on a small screen of a

PDA effective, so maintaining a high level of usability even for users which are mobile. As it is possible to note from Figure 2, the precision requested for selecting the different areas of the graphs does not pose strict constraints to users on the go, as it is a fairly easy task selecting a bas on the bar chart even for a mobile controller. In addition, the most critical information is visualised in various redundant ways: it is not only displayed on the graph, but also emphasised by the fisheye view, and further displayed in a dedicated tool-tip activated on the window.

Conclusions

In this paper we have presented a method composed of two main phases: a distributed task performance analysis, which aims to identify potential safety-critical issues through the analysis of deviations from the task plan and the information necessary for its accomplishment; and a transformation-based tool able to take the result of an envisioned conceptual model and obtain interfaces effective for the activities to support. One key advantage of this method is the possibility to support design in safety-critical contexts when the introduction of new mobile technology is considered. This result is obtained because the analysis is able to consider the context and how it can affect the user interaction and the tool is able to generate interfaces that are able to adapt to the feature of the devices considered.

In this way the environment supports the work of multi-disciplinary groups where the result of the conceptual design can be used to actually support the development phase.

Future work will be dedicated to extending the multi-modal aspects of the interfaces generated by the tool.

References

Buisson, M., Yannick, J. (2001). *Design Issues in Distributed Interaction Supporting Tools: Mobile Devices in an ATC Working Position*, Proceedings of Mobile HCI 2001.

Fields, R.E., Wright, P.C., Marti, P. & Palmonari, M. (1998). *Air Traffic Control as a Distributed Cognitive System: a Study of External Representation*. Proceedings of the 9th European Conference on Cognitive Ergonomics - ECCE-9, EACE Press.

Fields, R., Paternò, F., Santoro, C., Tahmassebi, S.(1999). *A Method for the Comparison of Design Options for Allocating Communication Media in a Cooperative and Safety-Critical Context*. ACM Transactions in Computer-Human Interaction Vol.6, N.4, pp.370-398, ACM Press, December 1999.

Hollan, J., Hutchins, E. & Kirsch, D. (2000). *Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research*, in ACM Transactions on Computer-Human Interaction, 7 (2), p. 174-196.

Hutchins, E. (1995) How a Cockpit Remembers Its Speeds. Cognitive Science, 19, pp. 265-288.

Mori, G., Paternò, F., Santoro, C. (2004). *Design and Development of Multi-Device User Interfaces through Multiple Logical Descriptions*, IEEE Transactions on Software Engineering, August 2004, Vol.30, N.8, pp.507-520, IEEE Press.

Paternò, F, Santoro, C. (2002). *Preventing User Errors by Systematic Analysis of Deviations from the System Task Model*. International Journal Human-Computer Studies, Elsevier Science, Vol.56, N.2, pp. 225-245.

Paternò, F. (1999). Model-based Design and Evaluation of Interactive Applications, Springer Verlag, ISBN 1-85233-155-0, 1999.

Wilson, S., Johnson, P., Kelly, C., Cunningham, J. and Markopoulos, P. (1993). *Beyond Hacking: A Model-based Approach to User Interface Design*. Proceedings of HCI'93. pp.40-48, Cambridge University Press, 1993.