Improving Dependability through a Deviation Analysis on Distributed Tasks in Safety Critical Systems

Ana-Maria-Marhan[1], Fabio Paternò[2], Carmen Santoro[2]
[1] *Institute for Educational Sciences, Bucharest (Romania)*
[2] *ISTI-CNR, Pisa (Italy)*
{Anamaria.Marhan, Fabio.Paterno, Carmen.Santoro}@isti.cnr.it

Abstract

This paper describes a method aiming to support dependability in interactive-safety critical systems. It proposes a multidisciplinary approach integrating areas usually considered separately -task modelling and distributed cognition- in order to analyse if possible deviations occurring in the performance of a system might affect its overall dependability. The basic idea is that analysis of interactive systems requires a clear understanding of the information needed to accomplish the tasks, and this information has to be derived from both internal cognitive representations and external representations provided by various types of artefacts. We illustrate our approach also applying it to a case study in the air traffic control domain.

1. Introduction

System dependability is generally defined as *a property of a system such that reliance can justifiably be placed on the service it delivers* [5], so its objective is ensuring good level of confidence that potential failures might not lead to hazardous consequences. In this work we propose a systematic method to assist designers in the analysis of interactive systems through an analysis of deviations –based method developed in a previous work [9], and here extended to provide more explicit consideration of the distributed cognitive resources supporting task accomplishment.

The basic idea underlying this work is that a failure of task performance is the consequence of an *inadequate access* to the distributed information resources supporting task accomplishment [4, 11]. Hence, our particular objectives are:

- To propose a design method that systematically analyses task accomplishment, detects potential deviations, and provides design criteria grounded on distributed cognition analysis;
- To support designers to better analyse the impact of introducing new technologies and the potential implications in terms of user support.

In the following sections, after briefly analysing the state of the art in the field we present our approach and introduce the case study (in the air traffic control domain) that will be considered. Then, we select one scenario in which a mobile device is used to carry out specific domain tasks and on which the application of the method will be illustrated. Finally, some conclusions are formulated.

2. Related work

One of the challenges currently faced by HCI analysis and design is that the supposed interaction device is no longer confined to the desktop, but reaches a complex network world of information and computer-mediated interactions. Distributed Cognition [4] may offer the intellectual basis for a paradigm shift in thinking about information-based work activities by analyzing how new systems fit into (or disrupt) the co-ordination of existing work practices and how the variety of current media for information representation may affect task accomplishment.

Nevertheless, the main limitation of the approaches based on DC paradigm is that, so far, they do not provide systematic support to designers who often confide in their personal intuition to translate some general principles in specific design criteria to be applied to the case study under consideration.

Therefore, in this paper we will focus on suggesting a possible way for integrating a systematic task-based modelling approach to HCI design and evaluation, with new analysis criteria derived from a DC framework. In this respect, we turned our attention to task-based modelling approaches and their supporting tools.

For instance, CTTE [7] provides an integrated set of features to represent and analyse task models, and it may provide valuable support for designers in

analysing potential or real deviations from the current task model, assessing their impact and consequences on the overall system dependability, and proposing alternative design solutions.

3. A method for design and evaluation of safety-critical systems

As illustrated in [9], a deviation analysis will usually follow three main steps:

- *Development of the task model of the application considered*. In this phase, the current system design is analysed and specific task performance requirements are identified. ConcurTaskTrees (CTT) notation [10] may be used to provide a description of the tasks that have to be performed including their temporal relationships, objects manipulated and task attributes.
- *Analysis of deviations related to the basic tasks.* The basic tasks are the leaves in the hierarchical task model, tasks that the designer deems should be considered as units of analysis.
- *Analysis of deviations in higher-level tasks*. This phase allows designer to identify *group of tasks,* and consequently, to analyse deviations that involve more than one basic task (i.e., analysing if the considered group of tasks are accomplished following the prescribed ordering).

CTT task models are structured in a hierarchical manner that allows an analysis at various levels of abstractions. At the lowest level, CTT supports a detailed description of the information objects manipulated by a task, which are classified in: *perceivable* (information directly perceived at the user-interface level), *cognitive* (cognitive representations or user-embedded information), and *domain* objects (entities that are internally represented and manipulated by the system). Similarly, tasks are classified depending on their performance allocation: *user tasks* require an internal (cognitive) performance, *interactive tasks* implies a direct action of the user upon the external world, *system tasks* are completely automated tasks. While deviation analysis has proven to be useful for generating valuable design recommendations, we realised that it can be increased in order to support the analysis of the current (or envisioned) system of distributed information representations, identify possible sources of failure, and alternative design options.

3.1. Integrating CTT task modelling and DC

The evaluation method we propose is based on the idea that a criticality in task performance is a consequence of an inadequate access to the information distributed among the interacting components (i.e. human and computer, human and others through the computer, etc.).

Based on the CTT description of the (real or envisioned) *plan of tasks* that the users have to follow in order to achieve their goals, the analyst will identify potential points of criticality in the task model, and analyse the information supporting task accomplishment, its representation and distribution across the task space. More in detail *the basic steps* of the method are*:*

- First, critical tasks are identified (they can be basic tasks or group of tasks);
- Then, the distributed representations supporting task performance are identified and described;
- Potential safety-critical deviations for each task are considered;
- And finally, alternative design solutions implying less safety-critical interactions are identified and evaluated.

A set of heuristics extracted from the DC literature orients the analyst in analysing the properties of representational elements distributed across the task space, and systematically reasoning about alternative ways of representation of information distribution supporting task accomplishment.

3.2. Analysis steps

When the method is applied three aspects need to be analysed: task properties, information representations, and possible deviations.

3.2.1. Defining task properties; CTTE supports identification of several task properties that could impact upon system safety and usability:

a. *Task category and type:* Depending on task category (i.e., interactive task), different types of task are meaningful (monitoring, control, editing, etc.).

b. *Task platform:* It refers to the physical environment (i.e., interaction devices) that embeds the representation of information.

c. *Task frequency*: It is one of the parameters that may affect the overall evaluation of the level of 'criticality' or 'importance' of a task.

d. *Other properties (that could affect task 'criticality')*: cooperation involved in the task, need of real time performance/responses (task urgency), strong dependency with other tasks.

3.2.2. Analysis of the representations associated with a task. This step identifies the representations

supporting task performance (i.e. objects manipulated by tasks and their representation). A number of attributes describing the relevant representation properties have been identified:

1. A*ccessibility*: refers to the type of *access* (i.e., *sequential* or *concurrent*) allowed to a representation available in the task space. The sequential or concurrent access to information depends on its representation form and implementation modality (i.e., audio or visual), and on the type of platform used.

2. Observability: refers to the user's ability to view components of a representation and relationships between components easily. For instance, is the representation a shared resource that the entire community can draw from - i.e. a map or a script? Or it is an information only locally available to the controller annotating strip on his PDAs? From this point of view, the visibility of a certain information may be: *local* to individuals; *shared* (e.g. by the members of a team); or *globally available* to all.

3. Persistence: the information represented in the task space may vary in persistence from permanent (i.e. strips) to relatively transient (i.e. vocal communications) [3].

4. *Flexibility of change*: refers to the possibility to autonomously update and modify a representation so that to reflect changes in the system. Some representation media ability to change *autonomously* to reflect changes in the system (i.e., radar),while others might be characterized by a certain resistance to change (maps, procedures, etc).

5. *Cognitive tracing and interactivity:* refers to the extend in which a person is allow to annotate and update an external representation, leave cognitive traces, i.e. mark, update, and highlight information (i.e. paper strips).

6. *Translatability:* refers to the ease of transferability of an information to other representation forms or storage media (for instance, a digital value can easily be concerted to a verbal form by reading; an analogue description is less easily propagated); re-representation allows reconfiguring and multiple views of information.

7. Integrability: refers to the possibility to integrate multiple representations available across the task space (i.e., visual with verbal channel, from various locations, etc.) into a single co-ordinated representation. Specific properties may facilitate (or increase the difficulty of) this process: *combinability* and *comparability* with other representations available in the user's task space, and *mobility* (possibility to resize/move/re-arrange items in the task space).

3.2.3. Analyse deviations from the task plan – In this phase the analysis evaluates how the current configuration of representations is effective for the task considered (and in the way this task is supposed to be carried out in the considered system). For example, if a task is a frequent monitoring task and in the considered system the representation of the current situation cannot easily be *compared* with the expected situation, this could put a heavy (and risky) workload on the controller who continuously has to accommodate this information before actually using it.

More in detail, our evaluation will be driven by means of a number of *guidewords*:

a. *None*: the representational element supporting task performance is not available in the task space, or not visible, or not observable (or a combination of them).
b. *Other than*: less, more or different representational properties than required is provided by the considered representation.
c. *Wrong timing*: the resources required is available, but either later than required, or earlier than expected.

For each task, the results of the analysis may be stored in a table with the following information:

- *Task*, indicating the task currently analysed, together with some properties relevant to our analysis;
- *Information distribution,* indicating the information supporting task performance and its representation;
- *Guideword*, indicating the type of interaction failure considered;
- *Explanation,* explaining how an interaction failure has been interpreted for that task;
- *Causes*, indicating the potential causes for the interaction failure considered and which configuration of resources might have generated the problem;
- *Consequences*, indicating the possible effects of the occurrence of the interaction failure in the system;
- *Recommendations,* providing suggestions for an alternative distribution of resources, if any, able to better cope with the considered interaction failure, through different means: preventing the deviation, suggesting compensating actions, depending on the concerned level of criticality.

Thus, not only the evaluation has to consider if a different allocation of resources may be envisaged, but also if and how different representational forms and media may result in a significant improvement for the overall system's safety and usability. It is worth pointing out the strong, multidisciplinary effort that this evaluation requires in order to be carried out effectively and provide significant results.

4. The case study

In order to show an application of our method we consider a case study related to a real setting: the Rome-Ciampino ACC (Area Control Centre). We visited the centre and interviewed a number of controllers working in it. It hosts en-route controllers' teams plus an approach working position that has to control and sequence the aircraft access to the runways of the close major Fiumicino airport.

During the en-route phase aircraft are managed by two air traffic controllers (a couple for every region or *sector* in the airspace) equipped with various screens with a graphical visualisation of the current situation in the sector: the *executive* controller maintains continuous contact with aircraft using the VHF radio and headphones and is directly responsible for maintaining the appropriate separation distance between aircraft; the *strategic or planning* controller basically performs medium-long term planning (identify future conflicts, planning future traffic), decides on flight separation and co-ordinates with strategic controllers of adjacent sectors. Other roles are present: one *chief* (who has a disciplinary role), one *technician* supervisor (in charge of monitoring the overall instruments' functioning from a technical point of view), one *flow controller* (who accesses estimated information related to the flight in the next hours) and three (or more) *supervisors*, in charge 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 handling personnel resources.

5. Applying the method to the case study

In order to show how to apply our method we consider a scenario related to the Ciampino case study.

5.1. Scenario: take over between two controller teams (approach position)

Description of the current situation: During the working day a number of shifts and handovers of positions (when a different team of controllers prepares to take-over responsibility of the sector) occur within the control room. The handover requires the new team to access the traffic context (i.e., observing a few minutes the traffic, hearing communications between controllers in charge, looking at the paper strips and even asking questions for getting clarification) in order to build an accurate mental picture of the current situation and become aware of any previous traffic events that are likely to impact on future events. However, verbal communications, that are normally used in enroute working positions, cannot be used within the APProach suite (APP), the busiest one in the control room as it requires a lot of coordination with other controllers and pilots and very short time for planning (and implementing) strategies. It is not by chance that this position is the only one in which controllers still use *paper* strips instead of *electronic* strips since they allow the fastest and most natural interaction (manual annotations), and are invaluable for constructing a shared situation awareness (i.e., they are generally displayed in a rack between the controllers, allowing them to have visual access to this information).

The hypothetical scenario: In the envisioned system during the 'overlapping' period the new team is supposed to be provided with a PDA, a device suitable for storing history and context and enable customisation of representations, allowing reconstructing the current situation while reducing to a minimum the need of verbal communications between the two teams.

5.2. Description of the task model

According to the task model shown in Figure 1 the 'new' controller has to understand what is currently going on at his working position by gathering relevant information from the information sources available in the task space (paper strips, radar, PDA, other controllers, etc.) so as to update his mental picture of the air space situation until actually "take over". Hence, during such time interval (right before the replacement of controller on-position until starting to act upon the real system having full control on the ATC equipment), the PDA may support the access to critical information. If we consider the parent task as a whole, it becomes apparent a complex configuration of representations supporting task: i.e., paper strips and vocal communications exchanged in the close proximity of the suite (which are globally shared by all the teams); information on the PDA (visible only to the non-operative controllers who will manage the traffic in the next future), etc.

After taking over, the PDA is supposed not to be available anymore to the operative controller.. An interesting issue raised by this scenario is the switch from non-operative to operative role, and the accompanying change of the work context in terms of tools, media and platforms used.

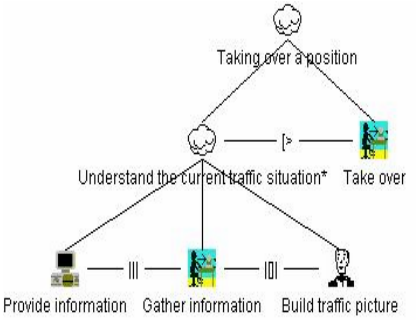

Figure 1: The task of taking over a position

Role: Approach controller taking over a position
Task: Taking over a position (replace the controller in charge and start to actually control the system).
Task properties:
a. *Task platforms:* PDA, paper strips, screens (with related tools), telephone, headphone, but also user-embedded or cognitive representaions acquired during overlapping period (by hearing communications, interacting with their PDA, and eventually asking questions to controllers in charge).
b. *Task frequency*: medium
c. Other properties with impact on task 'criticality': need of starting immediately to operate in the real system (e.g.: respond to pilots) as soon as take over is completed.
Analysis of the representations associated with a task.
a. Object supporting task:

- Mental picture of the current traffic situation (built within the overlapping period by hearing communications in the ATC environment, interacting with the PDA, asking questions to controllers in charge, etc.);
- Graphical representation on the screens (distance might prevent the non-operative team to access all the data)
- Audio data coming from headphones

b. Availability: both *external* representations (graphical: on the screens; audio: headphones) and *internal* representation (mental picture) of the situation.

- *Platforms* (or interaction tool supporting access): screens, headphones, pda (only available before switching)
- *Accessibility*: concurrent access to audio and graphical information (audio/graphical data continuously compared to the information maintained in controller's mental picture). Sequential access to verbal data from headphones.

c. Visibility: Data from headphones and PDA: Local; Information on screens: Shared by team members
c. *Persistence*: transient access to audio information;
d. *Flexibility* of cognitive tracing and interactivity: high
f. *Operations and actions supported*

- *Comparability* with other objects/representations available in the task space: Low. Representations on the screens not immediately comparable to that they received on the PDA (which they used during the overlapping period), because of diversities between the supporting platforms;
- *Combinability* (should allow user to select novel forms of combinations of information): Low. Controllers have to switch from using PDA to using huge screens; no combination is allowed.
- *Reconfiguring and multiple views:* Low. Every personalisation allowed on PDA (a device with limited capabilities and no possibility to update the real system) is banned on the fully operative large screens, in order to prevent the new team to have difficulties in interpreting views resulting from adaptation process.

Analysis of representations' configuration associated with the task
Guideword: Other than (while taking over the information provided is different from that expected)

- *Less:* the information is *less* than required
  - *Causes:* controllers have annotated some information on their PDA, and this device is supposed not to be used anymore in the fully operating system.
  - *Consequences*: possible overload on the memory of the controllers, who tries to recall their annotations and possible time wastage for rebuilding such information; possible distractions for the controller
  - *Recommendations:* enable controllers to have quickly available their annotations (also providing automatic deletion of 'obsolete' information) in the real system
- *More:* the information is *more* than expected
  - *Causes:* the fully operating system has to provide information on every aircraft (a/c) controlled, whereas during the overlapping period the controllers had put their attention just on some selected a/c (those with 'highest priority', according to some priority criteria)
  - *Consequences*: controller wastes time identifying concerned a/c
  - *Recommendations:* enable the controller to e.g. select an a/c on the PDA and make it highlighted on the huge screen (improve

combinability) and comparability (easiness in making associations between the two views)

- *Different:* the information provided is different from expected
  - *Causes:* the representations provided by the PDA in the overlapping period might have slightly biased the controller's mental picture against the representations available in the fully operative system with huge screens and headphones (the representations may strongly differ), if the two representations are not immediately comparable.
  - *Consequences:* overload on the controllers' memory (they try to suit their mental picture to the current situation as it is represented in the fully operative system)
  - *Recommendations:* provide mechanisms to "smoothly" move from one device/picture to another; consider the possibility of actively transferring information from PDA to real system plus the possibility of using PDA as a control device for the real system -although for a very short period of time.

The recommendations highlight under which conditions the use of a mobile device can provide useful support in the context considered.

6. Conclusions

Some authors have criticised task-based models of interaction, because they think that such approaches are not able to address the importance of context in interaction, and the distinction between tasks as described and tasks as observed in practice. A key concern in this case is the problem of characterising the context of action.

On the other hand, distributed cognition is particularly concerned with the context of work and the notion of distributed representational state, and the importance of mutual knowledge in guiding action. However, it is criticised for its high qualitative approach and difficulties of translating the results of DC analysis in the design practice.

A multidisciplinary integration of the two approaches could produce a mutual reinforcement at their both conceptual and methodological level and in this paper we have presented a method aiming at achieving such a goal. The method has been discussed within a case study in the air traffic control domain.

7. Acknowledgements

This work has been supported by the EU TMR ADVISES (http://www.dcs.gla.ac.uk/advises/). We also thank ENAV for allowing us to access the Ciampino Air Traffic Control Centre.

8. References

[1] Buisson, M., Yannick Jestin, Design issues in distributed interaction supporting tools: mobile devices in an ATC working position, Mobile HCI 2001.

[2] Fields, R.E., Wright, P.C., Marti, P. & Palmonari, M. (1998). Air traffic control as a distributed cognitive system: a study of external representation. Proc. of the 9th European Conference on Cognitive Ergonomics, EACE Press.

[3] 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, Dec. 1999.

[4] 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), pp. 174-196.

[5] Laprie J.-C. (Ed.), *Dependability Handbook*, LAAS Report n. 98-346, 1998.

[6] Mertz, C., Chatty, S. and Vinot, J.-L.. *The influence of design techniques on user interfaces: the DigiStrips experiment for air traffic control*. In Proceedings of HCI Aero IFIP 13.5, 2000.

[7] Mori, G., Paternò, F., Santoro, C. (2002). CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. *IEEE Transactions on Software Engineering*, pp. 797-813, August 2002 (Vol. 28, No. 8).

[8] Norman, D. A. (1988). The psychology of everyday things, Harper Collins: Basic Books.

[9] 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.

[10] Paternò, F. (2003), ConcurTaskTrees: an engineering notation for task models. In Dan Diaper & Neville A. Stanton (Eds.) *The Handbook of Task Analysis for Human Computer Interaction*. London: Lawrence Erlbaum Associates,483-503.

[11] Zhang, J., Norman, D., (1994) Representations in distributed cognitive tasks. Cognitive Science, 18(2), 87-122.