Bridging Software Evolution's Gap: The Multilayer Concept

Bruno Merlin[1], Christophe Hurter[2], and Mathieu Raynal[1]

[1] IRIT, UPS, 118 Route de Narbonne,
31062 Toulouse CEDEX, France
{merlin,raynal}@irit.fr
[2] DSNA/DTI R&D, 7, Avenue Edouard Belin,
31055 Toulouse CEDEX, France
hurter@cena.fr

Abstract. The multilayer interface concept is used to promote the universal usability, to smooth the transition to new systems and working methods and to help the user optimize his interface for the management of contextual situations. In this article, we will explain how this concept can help to tackle a serious issue for R&D projects: the integration of the innovative concepts into the operational environment. To illustrate this, we will explain how we used a multilayer interface to promote a way to integrate different concepts currently in maturation in the R&D sphere.

Keywords: Multi-layer interface, direct manipulation, working method evolution.

1 Introduction

The operational and technological concepts embedded in a system are limited by the state of knowledge at the time of the system specification. Thus, the time span for implementation and deployment is long, given that a system is generally a generation late compared to the current technologies. In the field of air traffic control, where the major systems have at least twenty years life expectancy (notably due to the intrinsic cost of their development and their inferred costs: user and maintenance training), important developments are conducted only with real technological leaps. In this context, working methods and working tools have to be reconsidered. However, during the life of a system, its changes are minor and generally this system does not implement the research yield. An end-of-life system is therefore 20 to 30 years late compared to the research field.

In order to meet the constraints of traffic incensement, the challenge of the next system is to provide a greater responsiveness in its ability to integrate the product of innovation. New systems should be able to quickly implement new tools in the air traffic controller environment. However, they must also guide controllers through the transformation of their working methods.

The goal of our approach is to show how the multi-layer interface which is HCI design paradigm can solve a double issue: the multilayer interfaces are able to smooth

M. Kurosu (Ed.): Human Centered Design, HCII 2009, LNCS 5619, pp. 266–275, 2009.

out the transition towards new working methods, and they can also be an architectural support for the management of a system's roadmap.

Through the analysis of different research projects, this article illustrates the main issue that faces research projects in order to be integrated in the controller context. Then, after introducing the concept of multilayer interface, we will show how these projects could use the multilayer concept to improve new tools and working methods integration.

2 From the Concept to the Operational Environment

The research and development yield is often awkward to implement in the operational industrial field, especially for critical activity such as nuclear power, aviation and air traffic control. Several projects, even advanced ones, struggle to reach the operational environment. To illustrate and to find out the reason of this difficulty, we will review several examples from the Air Traffic Control field.

2.1 The Appearance of the Color Screen and the Mouse

Even if the color screen technology was from ages integrated in our daily life since decades, it took years to expand them in air traffic controller working positions. Many stakeholders agreed to merge the color technology with the radar screen, but many factors had slowed down this migration. The first barriers were financial due to hardware costs, development and to technology's certification.

But, the main obstacle was the impact of the new representation design on working methods. With due reason, many controllers were worried with the new design use. The color was used to filter the radar screen by segregating information between important, secondary (shaded flights) and insignificant (hidden). This filtering was supposed to reduce the cognitive workload of controller by highlighting the main data. But the controller had difficulties to delegate this filtering to the system. They feared being induced in error and could not easily remain fully confident with the new design.

Long training periods smoothed out the brutal transition between monochromatic and colored screen. The same technological issues happened with the mouse advent.

2.2 Conflict Detection Software Issues

Medium Term Conflict Detection (MTCD) is a set of tools to assist air traffic controllers to forecast potential conflicts. Their main features are: aircraft and altitude screen filtering, supervision of relevant flight parameters and decision planning with time line tools.

Despite the significant potential brought along by such tools, their integration in the operational environment and in the current working methods raises several problems. HCI constraints, closed systems, formation costs and adaptation of controllers to new working methods make this step awkward.

2.3 ASTER Issues

The ASTER project [2, 5] focuses on the use of a vertical view in order to improve traffic management in the airport approaches (ETMA). This tool provides an electronic stripping [9, 12] using a display which gives the vertical position of the aircraft. This view facilitates the management of transverse flights in the main flow of arrivals and departures, in order to reduce the number of flight level instructions (less level instructions yield fuel savings, among other), improve the quality of traffic management in general.

Early in the design process, the issue of integration into the current control working position and in the working methods arose and became one of the main challenges. The vertical projection of traffic, proposed by ASTER, belongs to a family of electronic stripping tools. These tools aim at replacing the use of current paper strips. Air traffic controllers were involved in the user-centered design process [4]. They are offered the possibility to use these electronic strips in the same way as they did with paper strips. In addition, new design tools can offer them an opportunity to switch to new working methods.

In the light of difficulties faced by other projects, the only viable way to envision this project in terms of integration lead to the decision of developing independent tool taking the place of the existing paper strip board. This tool had to offer a consistent set of features and yet minimize interaction with the existing systems. Furthermore, it could not become an additional interaction tool redundant with existing tools or existing paradigms. The interaction with other separate tools (i.e. radar image or sequencing tools) was allowed only to provide optional features, but could not be a prerequisite for the operational viability of the system.

2.4 Results

The transition from a Research and Development (R&D) project to an operational tool is generally difficult. The projects, often forced to make sacrifices in order to fit into the existing environment, are weakened. Several factors can, however, facilitate the integration of innovative products: Structural improvements in the design and in the interactive software architecture for a greater flexibility of integration of new tools; Develop on-going training to assist users with the evolution of their working methods; Allow provision for the optimization of visual and interactive interfaces to include the handling of non nominal situations.

3 Multi-Layer Interface

The concept of multilayer interfaces (MLI), proposed by Ben Shneiderman [8, 13], aimed at promoting a more universal usage of applications. Later, the concept was extended to offer two additional services: Involve interactive systems in a continuous evolution process [10]; Adapt the software in order to be more efficient through a range of different tasks [5]. These different concepts allow us to identify where the multilayer's interfaces could also support the management of a long reaching project.

3.1 Universal Use Paradigm

The multi-layer interfaces were initially designed to promote universal use of application and allow various users (novice, amateur, expert) to use the interface efficiently, despite having heterogeneous objectives and training levels. These interfaces allow different types of uses [13] (from the most superficial to the most complex use), by activating [8] or refining the use of functions [5] and adapting visual density to the user's skills [6].

A group of functions and the corresponding visual entities define a layer. Transitions between active layers can be controlled either by the user or by the system. When layer selection is automatic, the selection is based on an analysis of the user's activity [5]. The multi-layer interface tries to help the user to gradually improve his efficiency with the software while retaining continuous control of it. To achieve this goal, we have to define guidelines for the creation of each layer.

3.2 Supporting the User along the Tool Evolutions

The concept of MLI, proposed by Schneiderman, has been extended in order to be used for the mutation of operational systems [10, 11], especially in the context of critical activity.

The new paradigm focuses on two additional issues: How to reduce the training period and to increase the application's acceptance during the transition between two systems; How to avoid brutal changes in the evolution of the applications. This property of multi-layer interfaces is achieved whenever several guide lines are followed to design the layers.

Design of Layer 1. The goal is to build the interface in a continuously evolving process and to avoid gaps between the old and the new software functions. New tools cannot suddenly challenge the working methods of the user. The interface must guide him. New tools must seduce him and progressively change his way of working.

The guide line therefore suggests establishing the old system (or at least a set of similar modality of services and interactions), as layer '1' of the new interface (see figure 1).

The direct consequence of this "conservatism" in layer 1 is that the user immediately finds a familiar environment, ideally completely similar. The training period becomes radically shortened because the user is already "layer 1" proficient. Using other layers can be done later, progressively and immediate mastery is not mandatory to use the new software.

Next layers design. The seconds guide line proposes to improve the interface in the next layers and yet to preserve an interaction redundancy with other layers. A new layer will be materialized with new functions and with new visual entities. This new layer will lead the user to consider new working methods. With these layers, the user may grasp new way to organize his workspace and new interaction paradigms that improve the accomplishment of his task.

Active layer selection. Whenever software deals with a critical activity, the user should always keep control over the system. He must be able to visualize, at any time, the active layer and be free to interact with it. This point is very important, and will help the user feel comfortable to explore new functions. At any time, the user must be left free to restore a familiar layer; bringing upfront a completely mastered interface. This provides him with the proper conditions to cope with a 'crisis' situation, at least until he feels he has got the same level of confidence with new richer layers.

Thus, the transition between two layers can be done by the user whenever he wants to, and in real-time, while using the interface. The transition must be reversible and triggered easily. The animation must be quick in case of stressful or heavy workload context. It is to be noted that this guideline clearly dismisses an automatic system which may choose to activate a layer. This strongly precludes unexpected transition in heavy workload situations.

User evolution. Suggested pattern of user progression is depicted in figure 2. Initially the user exploits lower layers (with mastered functions), and afterwards his range of used layers will slide to upper layers. The progression period of the user may vary with his assurance and his activity workload. In addition, the user's progress can accelerate with his curiosity and motivation.

While the original goal of the multi-layer interface was to encourage a heterogeneous population to use single software and to improve their skills, the goal here is to incite a homogeneous population to adopt a new working environment and new working methods at heterogeneous rhythms. The originality of the method is that it involves the user in a continuous design process of the interface. This tries to achieve the invisible integration of new tools.

3.3 Contextual Help

The first goal of the multilayer interface is to adapt the interface to a heterogeneous group of users. Its second goal is to support the user along software evolutions. A third functionality is to allow an application to optimize itself for the management of a specific task [6].

In this context, the MLI combines many different types of uses and therefore must adapt software in order to: Fulfill likeness functions with different layers; Allow users to deal more effectively with uncommon situations. The user can then optimize his interface according to his current activity.

The likeness functions context. As an example, the 'Eclipse' Integrated Development Environment (IDE) has characteristics akin to an MLI. This IDE proposes a common set of functions which help the programmer, regardless of what his programming language may be. The main menu is the same for a JAVA, C++ or UML developer. Independently of the way to use this IDE, Eclipse is identified as single software.

However, the views and some features are optimized to manage the specificity of each programming language and to back the programmer with appropriate support. The proposed functionalities (menus, buttons, tools…) are filtered to be coherent with the user activity. The layout is thus transformed to ease its exploitation.

Uncommon situation adaptation. Once again, we consider the example of software development activity. Whatever the programming language, the programmer activity is centered on two main tasks: The code edition; The execution, test and correction of the program.

To edit the code, the programmer needs to easily and quickly navigate, to be able to perform research on functions names or variables. While performing program correction and test phases, the user must be able to control the software execution.

An IDE, such as Eclipse, is able to adapt its functionality and the data it displays, and to switch between modes: navigation, edition and execution. In other words, this tool is contextualized in order to better manage momentary activity, even if the main activity is code edition.

3.4 Summary

The MLI offers a structural solution to address the issue of transition and implementation of an R&D concept. It allows: To integrate new tools with a structural theory; To offer a solution to help the user with the mutation of his working methods; To adapt a software with new tools and new visualization in the event of a momentary specific context.

Adversely, if we organized, in a particular layer, all the functionalities and all the visual entities of a given software version, the MLI can be a structural way to phase and schedule an interactive system. In other words, if each layer corresponds to a software version, the MLI becomes an architectural instrument to structure a project in the long term.

The user can quickly change his active layer. This possibility allows him to restore a better mastered layer, and then comfort his confidence with new tools in his working environment.

4 Concept Illustration

The opportunity offered by the multilayer interfaces is illustrated with the ASTER project [7] for Air Traffic Control. The MLI concept is here applied to an electronic stripping tool and serves to gradually build and design an Air Traffic Controller working position.

4.1 Electronic Stripping Features

The air traffic controller working position is intended for two collaborating persons. Each controller (radar and tactic controller) interacts with a specific part of the software application. Two radar screens, which display the current aircraft positions, are available. A strip is a standardized piece of paper used by air traffic controllers to managed information about aircraft. This abstraction of a flight is a support for coordination and communication between controllers. The strip allows specific features: Flight integration, marking, organizing and sharing aircraft instructions, work load planning, conflict avoidance, etc. (cf. videos available here http://perso.tls.cena.fr/acropole/2-1-eng.html).

The initial electronic environment is an electronic stripping (cf. figure 3) based on the DigiStrips experience [12]. This tool attempts to transpose accurately and with no modification the current working methods (for en-route and terminal sectors) into an electronic environment. All these features constitute the body of layer 1.

4.2 Working Method Transition

ASTER's initial goal was to re-organize the working method with the support of a vertical projection of the traffic. The idea was to complete the information brought by the main radar image. During the first two phases of experimentation [3, 4], despite the effectiveness of this tools (statistically assessed), the air traffic controller remained reluctant to go along with these new working methods.

The multilayer interfaces offered a solution, because they provided a smooth transition in working methods. Layers 1 to 4 have been designed to provide a smooth and continuous transition between consecutive layers. Thus, the user may work with a completely mastered environment and control his progress (cf. [10]).

4.3 New Tools and Concepts Integration

The second aim of the ASTER project is to show how the electronic stripping can be a transition step towards the future working position. It illustrates how the multilayer design paradigm could help new concepts to be integrated within an electronic environment for Air Traffic Control (ATC).

Each layer (layer 5 to 9) contains a specific tool corresponding to a new concept. Air traffic controllers may use this tool as contextual assistance. In a given context, air traffic controller is able to activate the specific layer which provides the needed functionalities.

Advanced planning tool & AMAN integration. Layer 5 is an advanced planning tool (cf. figure 5). The core of the tool is an interactive timeline (agenda), where a set of events is managed. These events can represent operational problems or advice (future conflict, catch up, flight descent etc.). The events can be created by a user or generated by different external sources such as algorithm treatments.

Quick and simple interaction enables the user to easily edit, move or delete part or all of existing events. Some urgent or critical events may trigger alarms or warnings if they have not been treated within a given time span.

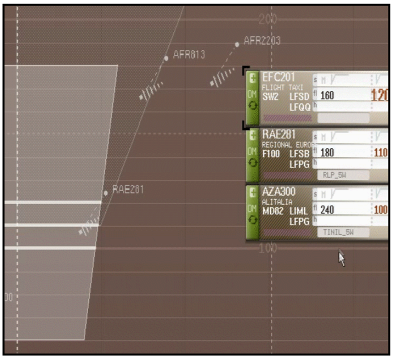
Event from MTCD

Event edition

Creation of user events & AMAN information

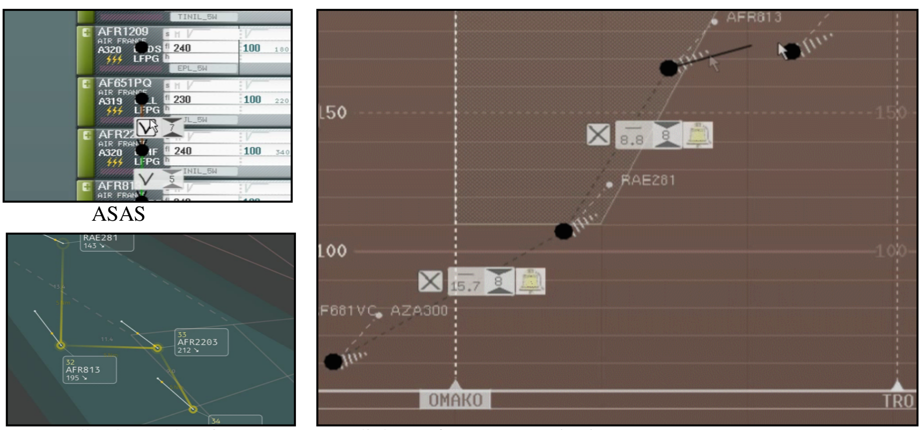
Fig. 1. Advanced planning tools

Layer 6 brings into the electronic strip system a complementary planning information generated by an arrival manager (green lines and dashes on figure 5). The left black columns of the timeline represent the expected sequence of arrival on two different runways (in our case, south and north runways at Roissy airport).



Holding management assistant. Layer 7 integrates a symbolic representation of the holding zone (the holdings are hippodrome areas where flights, separated in level, wait for descent when their destination airport is congested). by displaying the free and occupied flight levels. The representation facilitates the analysis of the planed vertical situation and enables to compare in real time the difference between the planed future situation and the actual current one. The system, informed by the controller through his management task, can identify and highlight inconsistent operations. For example: the lines highlighted in the holding area illustrate the future occupied levels. If any level is attributed to more than one flight by the controller, an alarm can be raised by the system.

Flight conformance & ASAS monitoring. Layer 8 enables controllers to delegate task to the system. At first, the controllers can program the system to monitor flight parameters (speed, level etc.) or given constraints. At the difference of the other alarms raised by the system, in this context, the controller requests explicitly to the system to the survey flight and to inform him in case of non-conform evolution of the situation. Moreover, the controllers can record an ASAS constraint (the respect of distance or a delay between flights) delegated to the pilot or to the system.



ASAS

Feedback at radar view

Flight conformance monitoring

Fig. 2. Flight conformace & ASAS monitoring

Data-link. The layer 9 integrates information and tools for data-link operation as well as a traffic monitoring system (data-link is an electronic message set of services between pilot and controllers and in the future between pilot and system or flight and system). The controllers can control the status of data link clearances and receive requests from pilots.

5 Conclusion

Our approach tried to take full advantage of the MLI concept and use it in the Air Traffic Control domain. We explained how it could ease the integration of R&D concepts and how it helped us to increase the acceptance of our product by smoothing the gap between old and new working methods. We applied and extended the multilayer interface introduced by Ben Shneiderman. Doing so, we noticed that this concept was efficient in the ATC field.

It is, however, difficult to estimate the benefits of the MLI independently from other design options applied to the tool. Lately, the tool was completely redesigned and the recent experiments are showing a better overall acceptability. But we have to acknowledge that it is difficult to measure accurately what is the relative impact of new design options, of new interaction paradigms or of the MLI on this acceptability. These different factors seem to be tightly interlinked.

In addition, the MLI philosophy implicitly guided our design choices in the sense that it forced us to: Think about the learning process in order to make it more natural and intuitive, and to render the tool itself self-explanatory; Capitalize on homogeneous and reused interaction modes [1]; And to make the tool more 'fancyful' to the user, because we had to induce him to explore the tool by himself,

We are therefore convinced that the MLI paradigm can prove a key element to simplify and accelerate the integration of innovation in an operational field. This is done by engaging the user, the working methods and the systems in a continuous mutation process.

References

1. Beaudouin-Lafon, B., Mackay, W.: Reification, polymorphism and reuse: three principles for designing visual interfaces. In: AVI 2000, Palermo, Italy (2000)
2. Benhacène, R.: A Vertical Image as a means to improve air traffic control in E-TMA. In: USA, DASC-Digital Avionics Systems Conferenc, Irvine-California (2002)
3. Benhacène, R.: As Rapid as PaperStrips? Evaluation of VertiDigi, a new control tool for Terminal Sectors. In: ATM 2005, Baltimore, USA (2005)
4. Benhacène, R., Merlin, B., Rousselle, M.P.: Tackling the problem of flight integration. In: ATM 2007, Barcelone (2007)
5. Clark, B., Matthews, J.: Deciding Layers: Adaptive Composition of Layers in a Multi-Layer User Interface. In: CHI 2005 (2005)
6. Gustavsson Christiernin, L.: Multi-Layered Design - Theoretical Framework and the Method in Practise. In: Winter Meeting 2005 Proceedings (2005)
7. `http://perso.tls.cena.fr/merlin/1-4-eng.html`

8. Kang, H., Plaisant, C., Shneiderman, B.: New approaches to help users get started with visual interfaces: multi-layered interfaces and integrated initial guidance. In: Proceedings of the 2003 annual national conference on Digital government research, Boston, MA (2003)
9. Mackay, W.E.: Is Paper Safer? The Role of Paper Flight Strips in Air Traffic Control. ACM Transactions on Computer-Human Interaction 6(4), 311–340 (2000)
10. Merlin, B., Hurter, C., Benhacène, R.: A solution to interface evolution issues: the multi-layer interface. In: CHI 2008, Florence, Italie (2008)
11. Merlin, B., Benhacène, R., Kapp, V.: Interface multi-layers et processus d'évolution des systèmes interactifs en activité critique. In: IHM 2007, Paris (2007)
12. Mertz, C., Chatty, S., Vinot, J.: The influence of design techniques on user interfaces: the DigiStrips experiment for air traffic control. In: HCI Aero (2000)
13. Shneiderman, B.: Promoting universal usability with multi-layer interface design. In: Proceedings of the 2003 conference on Universal usability, Vancouver, Canada (2003)