ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/250544818

Advanced decision aids for Air Traffic Controllers : Understanding different working methods from a cognitive point of view

Article

CITATIONS
6

READS
11

4 authors, including:



Raïlane Benhacene
Ecole Nationale de l'Aviation Civile
21 PUBLICATIONS **54** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



MoTa (Modern Taxiing) View project



ASTER (ASsistant for TERminal sectors) View project

All content following this page was uploaded by Raïlane Benhacene on 03 May 2017.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Advanced decision aids for Air Traffic Controllers : Understanding different working methods from a cognitive point of view

M.C. Bressolle*/**, R. Benhacene*, N. Boudes*/*** & R. Parise*/**

* Centre d'Etudes de la Navigation Aérienne, 7 Avenue E.Belin
BP4005 Toulouse cedex France
** Sofreavia, 3 Carrefour de Wieden, 92441 Issy-Les-Moulineaux France
*** Protectic, 20 Parc Burospace, 91573 Bièvres France

Abstract :

This paper presents some experimental results about a major air traffic controllers' cognitive process while working with a new decision support system called Erato. This project is developed by the French Research Center for Civil Aviation (CENA). We are aiming at getting a good level of understanding of how controllers use the main functions of the system, so as to ensure relevant improvements in the future. Results are organized in two sections. Firstly, we present some results about unexpected use of the system and of various working methods with Erato. Secondly, our objective is to show that this use of the system can be explained in part by cognitive processes that are carried out by controllers. The results presented concern default knowledge. We consider that such default knowledge constitute a cognitive trade-off and discuss their connection with representation updating.

1. Introduction

Air Traffic controllers have been confronted to a new decision support system (ERATO). Observations have shown that they developed different strategies for action to cope with the task complexity. The purpose of this article is to develop how, following a cognitive perspective, we have been able to understand their strategies and to account for mental mechanisms at work. This was done by the analyses of data collected during simulations with the Erato tool : linguistic queries, statistical and qualitative analysis performed on the data base thus created.

One salient characteristic of air traffic control is its complexity, which forces the controller into a trade off between the necessities of the task and his resources. We will show that a default knowledge mechanism is a means to put this tradeoff into practice, and permit an acceptable resource management.

2. The environment complexity

Air Traffic Control is often considered a complex task in terms of cognition. Its main aspects, according to Woods (1988) fall into three categories: environmental characteristics, characteristics of the operators, and those of the interfaces. In our opinion, the complexity is mainly driven by the coupling of the characteristics of the task (environment and interfaces) and those of the operator (Leplat, 1996). We can therefore expect that the introduction of a new tool for control will tend to shift or modify the initial complexity level. The way the operator handles the complexity will show through strategies for action carried out at the interface level.

The complexity due to the characteristics of the control task is generally represented by dynamism, indetermination, and the large amount of data to be processed at any time (for instance, there is over twenty basic data on one single flight plan). These characteristics induce inaccurate data, either because of its fuzziness or because of its uncertainty:

- data is fuzzy if the controller only has access to an estimate of its value. For example, the controller knows that an aircraft will fly over that fix at 12h, plus or minus 3 minutes.
- data is uncertain whenever its value is likely, but needs confirmation. For example, it is likely that a pilot requests to climb although his flight plan does not indicate this.

In addition to these characteristics of the environment, another difficulty emerges due to multiple sources of data, among which we can quote: pilots, interfaces, teammates, procedure manuals… Last, air traffic control is acknowledged as being a system at risk, be it in terms of accidents or in terms of the management of an individuals' cognitive resources (Amalberti, 1996) because, either for decision making or for effective action, strong timely pressures require that activity be synchronized to the environment's evolution.

Given the amount of information available and given the limits of man's perceptive and cognitive capacities, a filtering of all accessible information is necessary. However, if information can be overwhelming, it can also be missing with respect to some specific aircraft (temporarily or permanently), for example, when a track is not yet visible on the radar image.

In this environment, the controller puts into practice certain skills derived from his expertise, which

account for a cognitive trade-off in terms of resource management. This idea is developed below.

3. Building of a situation representation and task management: a cognitive trade-off.

In many instances, it appears that air traffic controllers' decisions are just as well motivated by minimizing their cognitive overload as by tactical reasons. This is typically the case of control actions developed when confronted with an otherwise uncertain diagnosis of conflict. This, in order to liberate cognitive resources that else would have been allocated to the monitoring of that situation (had the controller done nothing). This type of decision, obviously very context-dependent, accounts for a cognitive trade-off between the building of an accurate anticipation for a particular situation on one hand and the resources necessary for coping with the evolutions of the overall traffic on the other hand.

In Amalberti's model (1996), the cognitive trade-off reflects a constant risk taking to perform a task at an acceptable cognitive cost, this risk being accepted and monitored so as to minimize errors. Of this model, we shall retain that, in dynamic situations, the operator must adjust his level of understanding to the constraints imposed by action. More over, the moment of the action often requires that the operator be capable of taking decisions while monitoring his lack of comprehension, whether he may elect to act with missing or inaccurate, uncertain, fuzzy data…

A second dimension of the trade-off concerns the adaptability of task management to the context. In air traffic control, the complexity of task management increases with the introduction of new systems. This, because new tasks of interface management and consultation add up to traditional control tasks. And those too must be synchronized with traffic evolution. The use of new tools therefore requires that controllers integrate the interaction demands of the new system into their own cognitive resource management.

Underlying the cognitive trade-off described above is the availability of some cognitive skills. We now offer to describe those skills.

4. Default knowledge

Inspired by Reiter's model (1985) on reasoning by default in artificial intelligence, Leroux (1993) proposed an explanatory model for the way in which controllers treat situations although they logically lack of information to do so. Observations showed that controllers have, by experience, acquired knowledge about the customary evolutions of aircraft and that this knowledge is the grounds for an ability to anticipate future evolutions. This knowledge comes under the perception of regularities of the environment. For example, the repeated confrontation with a traffic on the same sector generates for the controller the emergence of waitings at various levels : expected behavior of the aircraft, traffic flow density for given routes and times of the day, destination of flights depending on the companies, pilots and adjacent sectors' probable requests …

Default knowledge is stored in the long-term memory and triggered, sometimes automatically, by the perception of specific stimulus (type of aircraft, hour, company…). This knowledge has the advantage of filling gaps in the data the controller handles, because a missing or incomplete information can lead to a reconstruction (or substitution) on the basis of a default reasoning, i.e. using exact rules most of the time.

If operators use cognitive skills, such as default knowledge, to face the complexity of the environment, they must also dedicate, as we said, part of their resources to understanding and managing the tools and interfaces they use. The amount of resources allocated to these tasks increases the more so when the logic of the system is not sufficiently mastered, or if the principles of the decision aid inadequately match the cognitive model of the operator.

As far as the ERATO tool is concerned, matching of the tool to the cognitive model was guided by the existence of default knowledge. For example, calculations of flight trajectory prediction have built-in margins to cover for variations induced by expert knowledge. Functions are aimed at reducing the controller's number of data acquisitions intended either to confirm the validity of default representations or, as will be shown later, to maintain situation awareness (SA). For example, the system permits to highlight certain parameters, typically when their values are not customary; or else to enable the controller to position geographic markers helping him to recall the monitoring of parameters (such as top of descent).

Nonetheless, Starter & Woods (1994) have shown that operators build a very fuzzy or incomplete model of how a complex system works, and that that model then impacts their utilization mode. By analogy with the type of knowledge involved in the anticipation of future traffic, we can imagine that the model of functioning relies on default knowledge. Gradually built by interacting with the tool, default knowledge would come out of the necessity to manage the interface in the absence of accurate data on the principles of its functioning. This default knowledge would be used to predict the system's behavior, mainly for what concerns the display of information and potentially their evolution through time.

This knowledge is justified by the complexity of the system and the controller's inevitably incomplete knowledge of the design logic. We can

expect that this knowledge starts to build up as early as the training period for the use of this tool.

Whatever the subject of default knowledge (traffic evolution, system's behavior, future activity of the teammates…) its efficiency holds to the fact that it covers the majority of cases of evolutions that are met, and that it costs little resources, especially when automated. However, it can prove risky if actual evolution does not follow the one which had been activated by default. Truly, the persistency of a default representation can be such that it can lead the controller to neglect certain information and make decisions incompatible with the actual evolution of an aircraft. Due to this, and to grant a complete efficiency of the default knowledge, the controller has to check its validity domain, and update its content, which brings us to our next item.

5. The validity of default knowledge

The monitoring of the validity conditions of default knowledge is necessary to grant its efficiency. This boils down to insuring that the actual situation conforms to what is expected in the default representation (for example : the aircraft evolves as expected, the adjacent sector transfers aircraft as thought…). These validity conditions rely on the value of some parameters of the situation , upon which a filtering retains only those essential, those that Leroux (1993) has called "sentry parameters". Any parameter can become a sentry parameter depending on the context in which the representation is activated (rate of climb, direction of the wind, arrival…).
The sentry parameters must remain inside a "normal" variation domain which corresponds to the validity conditions of the default representation. To control the validity of default representations is equivalent to maintaining a good Situation Awareness (SA) (Endsley, 1995). The author's definition is here given :
« Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future ».
The three stages of this definition can be transferred to our topic. One can expect that the first stage corresponds to the identification of sentry parameters. The second stage corresponds to the evaluation of a potential gap between the default value and the actual one for that parameter. Last, the third stage would consist of re-adjusting (up-dating) one's default representation according to the actual parameters of the situation.
For these reasons, as many authors have shown in actual or simulated conditions, some environments can generate a deficit of SA susceptible to jeopardize the safety of a system. Factors involved are mainly : the level of automation and lack of feed back (Endsley & Kiris, 1995), or an undue confidence in the system.
In air traffic, unless it goes unnoticed, loss of SA is characterized, for the controller, by the troublesome feeling of no longer being ahead of traffic, and no longer being in control.
In all cases, the level of SA does not seem to condition the degree of accuracy of the representations. Even under the hypothesis of a good SA, there is always room, for the controller, for an amount of risk taking, due to the characteristics of the data he processes and the default knowledge he performs.
To sum it up, the controller manages an environment whose complexity is largely due to the coupling of the type of data he processes and his degree of understanding of the system he uses. Confronted with this complexity, we consider that the controller's activity is subject to a cognitive trade-off between maintaining the accuracy of his mental representation and optimizing the system use in accordance to the cognitive resources he has available. We believe that default knowledge is a means by which the operator can regulate this cognitive trade-off. In this paper, our objective is to prove that this knowledge can just as well relate to the traffic evolution, the system behavior or the teammate's activity.

6. Global description of the Erato Environment

Erato (En Route Air Traffic Organizer) is a decision support system for en route air traffic control (ATC) which has been designed according to a cognitive engineering approach by a multi-disciplinary team (including engineers, cognitive ergonomists, air traffic controllers). The ERATO system software includes a set of decision aid tools designed for controllers (executive controller and planning controller). The HCI is composed of six area described as follows:
- The radar display: The main functionality of this window is the filtering function. This function presents, for a selected airplane, all aircraft, which are or may be in interaction (in conflict with it or constraining it). A filtering request highlights all the aircraft involved in potential interaction. This is designed to improve the detection of potential conflicts and ease representation updating. This window also includes several additional functions to enhance memorization and anticipation (extrapolation, geographical marker, clearance preparation, etc).
- The Reminder : The Reminder shows a dynamic view of problems organized as a time schedule, according to their urgency. The controller may adapt them to his/her needs by modifying their presentation. The problem tags presented by the system give a view of the future workload in order to help the controller manage his/her cognitive resources. It may also be helpful for representation

updating as well as for cooperation between controllers.
- The Table of Flights : This table presents the electronic flight strips corresponding to the airplanes involved in the sector.
- The Table of Filtered Flights : This is a display of the electronic strips of all flights involved in the filtering of a reference plane.
- The Flight Integration Window (FIW) : This window presents the callsign of new airplanes coming into the sector and enables to rapidly highlight them.
- The Aid for Exit (AFE) : This table of outgoing flights displays different lists of planes depending on the exit flows or beacons.

7. Method

A large set of experiments was conducted to assess the impact of the Erato environment on the activities performed by controllers.
Data was collected from 16 simulated traffic scenario using ERATO, lasting 90 minutes each. Eight pairs of controllers were involved, each composed of one executive controller and one planning controller. The controllers came from four different French en-route air traffic centers. They spent 2 weeks' training on the system : a first phase with classes and practice on the interface (three days) and a second phase to enable controllers to put the working method into practice with increasing traffic (six days).
For all the scenarios, the controlled traffic was picked from a real-time working situation and the level of traffic density was acknowledged by experts as heavy. In these dynamic simulations, all actions carried out and displayed on the interface were recorded. Thus, all system events (like the occurrence of a problem label, the display of an alarm, …) were stored in real time as well as any inputs made by the controllers (filtering requests, integration request, …). Verbal communication data collected (including VHF) amounted to about 20000 communications.
We defined a data coding system relevant to events generated by the system, actions on the system (consultation or input activities), and controller communications at the working position. The following data was gathered :
- events generated by the system itself corresponding the planning controller (PC) interface or the executive controller (EC) interface ;
- the actions undertaken by each controller (the PC and the EC) ;
- the verbal protocols recorded while controlling the scenario (thinking aloud, communications between the radar controller and the planning controller, between controllers and pilots or neighboring sectors) ;
- verbal protocols recorded during interviews (which lasted about 3 hours) just after the scenario session, based on a replay of an air traffic control simulation. The replay of the controllers' actions performed on the display was synchronized with videotape of controllers at work. This gave them the relevant communications and further data to retrieve the situational context. Both controllers were asked to give explanations about the actions taken along with conflict detection and resolution, to describe their analysis of the moment, their intentions and expected outcomes.

8. Results

8.1. Main results on the use of the system

These results come from Abdesslem and al., 1999. We shall here strive to highlight some of the main specifics of the interface utilization and the issues they bring about as to the mobilizing of cognitive resources and the enacting of underlying cognitive mechanisms. In particular, we shall focus on data acquisition behaviors and inputs that turn out to be different from what we reasonably expected (non updating of data, diversion from a recommended working method, etc.).

- Utilization of the system according to position occupied (EC/PC) :
First to be noted is the difference in behavior of the EC and PC as to data acquisition and acting upon the system. This difference shows up in various circumstances. For instance, on integrating[1] a flight (see Figure 1), the Flight Integration Window (FIW) is the preferred and main location for manipulation on the interface, especially for the PC. This, because the FIW is a part of the screen where a list is given of flight call-signs that have just entered the sector. For the EC, meanwhile, even though the FIW is more appropriate, action on the radar label is significantly predominant.

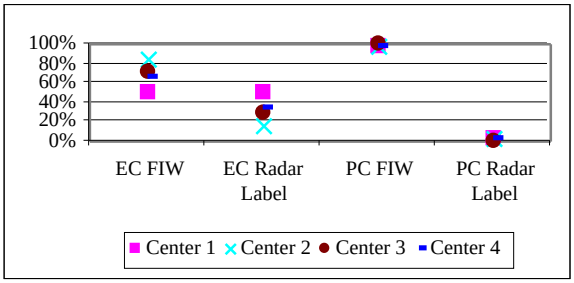


Figure 1 Location of flight integration per position (EC/PC) and for each control center.

We may believe that this choice of acting on the radar label is in relation with the "radar-oriented"

[1] Integration is defined as the moment when the controller takes hold of the flight and mentally incorporates it into the traffic.

activity of the EC. In other words, for him, against all odds, the detection of a flight not yet integrated will more naturally occur during a visual scanning on the radar image because this would not induce a perturbation of the activity during consultation as would have been the case with the FIW. Adversely, this latter data acquisition mode would be more adapted to the PC way of functioning. There would therefore be a trend to favor some information rather than other. However, integration at the radar level does not grant exhaustiveness, particularly when flights not yet visible[2] on the radar are concerned. The question remains as to why does the controller elect a means for data acquisition that does not grant him a complete representation of the situation.

In close relation to this differentiated choice of interaction mode on the interface, it has appeared that controllers may convey two types of control modes : a strip-oriented mode and a radar-oriented one. These two modes might well be accountable to the current situation in which controllers are confronted with two major tools for their work : the radar and the table of strips. Switching to the electronic environment of ERATO would not fundamentally question the maintaining of these two modes. However, we have remarked that some qualified strip-oriented controllers have difficulties in appropriating the four scattered representations that ERATO substitutes to the current table of strips (filtered strips[3], strips in the Aid for Exit[4], strips for the Table of Flights[5], and the Reminder[6]). But the difference between these two working modes also and mainly shows up when considering the use of functions.

If we move back to the description of the integration modes on the radar image, we may better understand the problem raised by the integration of flights not readily visible for some allegedly radar-oriented controllers. For those, the FIW forces too precocious an integration as the flight is not visible yet, and integration is thereby made with less efficient means, as the expected radar information is not available. Consequences on the utilization of functions is therefore expressed by a tendency to integrate those flights later than others, i.e. to wait until the radar track appears.

In relation to the EC/PC position of a team, the analysis of tool usage has revealed a strong correlation between the specificity of ERATO functions and the task sharing between the executive and the planning controller. If we refer to data regarding integration of significant flights, i.e. flights belonging to a problem detected by the Reminder, we note that they are more systematically integrated by the EC before consultation of the labels they belong to than can be seen for the PC. Thus, 72,6% of significant flights are integrated by the EC before consultation of the corresponding problem, versus only 57,9% by the PC. These results, reinforced by the amount of mere problem consultations in the Reminder, tend to show that the EC would rather be “flight-oriented” while the PC would rather be “problem-oriented”. The consequences in terms of useful information to be displayed are easily drawn and are currently reflected on the ERATO tool by a redundancy of flight parameters on interface representations. It would be appropriate, for training purposes, to anticipate such trends and therefore to provide controllers with tailor-made exercises, possibly adapted to their control mode in order to maximize the tool’s appropriation in both specific positions (radar or planning).

- Minimizing the number of consultation and input actions :

Still on this issue of optimizing the resources allocated for a task, evidences show that controllers, especially in an EC position, attempt to minimize the number of actions on the interface and noticeably the number of inputs. The alleged economy in the interface manipulation is backed by the assumption on controller’s capacity to build up a traffic representation based upon missing, inaccurate or even wrong data (Boudes and Cellier, 2000). This is an essential issue to understand the use of an interface, the more so when it needs not be manipulated to be used. We had noted this during flight integration phases when mental representation of a flight can be achieved without resorting to other functions such as filtering[7], the graphic route[8] or the filtered strips, i.e. without getting involved into a software integration. This functioning mode is plausible, given the fact that all information required for this construction is available else where and by reading on screen.

However, the most striking fact about this minimal utilization behavior on the interface concerns

2 Flight having not yet shown on the radar image and which are represented by complementary labels on the screen edges.

[3] Filtered strips are the computerised version of the paper strips of flights belonging to a filtering (see next page)

[4] Aid for Exit is a list of strips clustered by flows exiting the sector.

[5] The Table of Flights is a complete list of flights known to the sector, arranged according to their CFL.

[6] The Reminder is a list of problems detected by the system and listed chronologically in order of expected resolution time (ERT).

[7] Filtering is a function which permits to highlight all flights likely to interact (conflict or constraint) with a reference flight or problem as selected by the controller

[8] The graphic route is the display on screen of flight trajectory as defined in the flight plan

inputs. Having analyzed six hours of simulation recording, we were able to pick 235 flight level clearances and 224 route shortcut clearances. Figure 2 gives errors and mishaps (wrong first input, soon corrected), but also correct inputs that we were able to spot while trying to find coherence between input actions and verbal statements on the working position (calls to pilots).

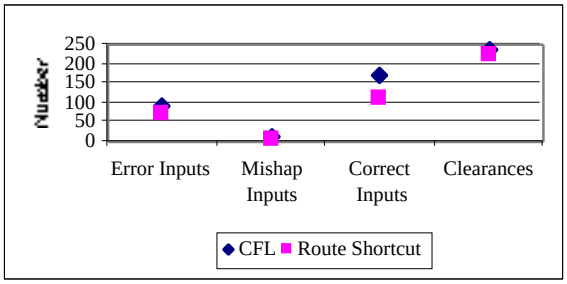


Figure 2 Layout of the types of inputs per theme (CFL or shortcut) - all positions considered.

Counting of effective clearances, as seen through controller inputs, reveal that only 168 CFL clearances were actually enacted on the interface, i.e. 71% ; as well as 114 inputs for route shortcuts, i.e. 51%.
These values can be analyzed from the mere point of view of input means for which some critical remarks have been made : too small input areas, erratic scrolling of menus, twofold inputs... But we should also look at the cognitive activity in order to find the cause for this absence of inputs. This, because it can come out of a deliberate decision of the controller to do without inputs that are of no added value or which appear to him to bear less risk than others. For instance, when the position of the aircraft on the radar screen tells him with no possible doubt that the aircraft is on direct route towards a known fix, making inputs into the system seems of less importance and are therefore naturally neglected.
These results brought us to draw a distinction between the use of an interface and its manipulation. This mainly translates into cautiously considering the possible ways of displaying useful information while avoiding to overload the interface and requiring too numerous manipulations. The Erato project has thus elected to multiply information available on screen by adding a second screen which will display the Table Of Flights (TOF) and the Aid For Exit (AFE). The Table Of Flights provides a comprehensive list of flights known to the sector in the shape of mini-strips arranged in order of cleared flight level. When enacting a filtering, it also gives the list of filtered flights by displaying full extent strips. The Aid For Exit organizes exiting flow per beacon by displaying the strips corresponding to the flights involved. This latter tool is of a considerable help for detection of conflicts occurring on flights exiting the sector. These two information displays complete the radar screen and the Reminder by creating specific location for data acquisition, sometimes redundant but differently organized in order to provide controllers with other types of aids, by minimizing the constraints of searching on the interface, therefore minimizing manipulation.

- Multiple consultations of Problem Tags :
We have also identified this minimized manipulation of the interface on the Reminder function. It appears that 31.2% of the problems displayed on the Reminder are not consulted by the planning controller. A detailed analysis of problems not consulted shows, amongst other, that this absence of consultation can read across with the very nature of information displayed on the radar label. This latter enables to detect a problem without having to go through the consultation of the Reminder label filtering. This behavior is further enforced by the characteristics of some Reminder problems (evolving flights) for which consultation of the filtering may be considered by some controllers as redundant with flights' filtering which can be performed on the radar label. Adversely, some problems consulted can be repeatedly manipulated by the controller for problem filtering. The figure below indicates the proportion of multiple problem consultation.

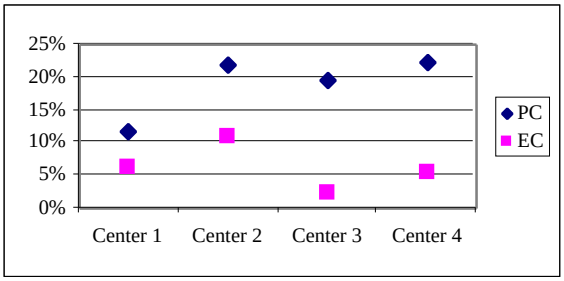


Figure 3 Proportion of problem consulted several times depending on controllers position and control centers.

We note that, more often than executive controllers, planning controllers made multiple consultations, and this in a noticeably higher proportion. These values are a clue to the added value of the Reminder in its function of supporting conflict marking. But this result, in conjunction with earlier expressed hypotheses on cognitive mechanisms developed by controllers, lead us to believe that those multiple consultations are the sign of representation updating. Representation updating is the ability for the controller to build a "temporary", evolving representation of traffic which will later be enriched, made more accurate, reviewed, confirmed or replaced. Multiple consultation of

flights or problems would be an observable aspect of that judgment.
In fact minimal usage of the Reminder by the executive controller and a multiple use of problems by the planning controller on the Reminder hint at the existence of cooperative modes inside a team. In particular, it seems that the low utilization of the Reminder by the executive controller is facilitated by the preparatory, anticipatory work of the planning controller. Aside from that, the recognition of this interface manipulation task sharing raises questions as to the nature of communication between controllers. In addition to verbal communication, information is forwarded by or via the system, and preferably triggered by either position, thus creating a "silent" cooperation. This analysis confirms the specialization by trade : radar or planning. Moreover, it is a guideline to interpret the task sharing utilization of the interface oriented towards an executive controller vs. planning controller specialization, and inducing a knowledge about the teammate's behavior which will orient choices of interaction with the system.

But the validation of these explanations and of the body of observations presented cannot go without a more extensive exploration of the cognitive mechanisms underlying these differentiated behaviors on the interface. It seems now worth verifying that any of the following : inputs not enacted, election of data to be acquired, ignorance of displayed values, decision making based on fuzzy or incomplete data, could partly be based on default knowledge.

8.2. Default knowledge

Default knowledge is a type of knowledge that is expressed by plausible inferences based upon incomplete data. We consider them incomplete either because they are not available (for example, the top of descent of a flight about to land) or because the controller does not use the information displayed by the system. This default knowledge is seldom verbalized by controllers and often remains implicit. To study it, we start by making systematic queries on linguistic markers followed by a qualitative analysis of self report verbalizations (15 interviews) in order to identify the different types of default knowledge at work (related to one's own activity, that of the teammate, of pilots and adjacent sectors, the system's behavior or traffic's evolution). This analysis is twofold. It is first based on a systematic search for verbalization characterizing a default knowledge, then further refined on the basis of a sample of terms that are clustered into classes.
These terms refer to the expression of knowledge acquired out of experience, habit, expectations, events that occur with certain frequency, that refer to what, in the current situation, happens on a daily basis, etc. We have grouped them in classes to make the results easier to interpret, according to the following principle : we have assembled terms that allude to the current situation (class 1), those relevant to events that occur generally (class 2), those having a certain frequency, which seem logical or coherent or are conditional (class 3), terms referring to habits, to routine or normal behavior (class 4) and terms relative to obligation (class 6). From there, a qualitative analysis of the interviews enabled us to confirm that these verbalizations actually correspond to the use of default knowledge ; and to select those that are relevant to this knowledge even though no marker is to be found in the verbalization (class 5).
The main themes corresponding to each class, in order of occurrence, are the following :
Class 1 : reality (60%), common (40%)
Class 2 : knowledge (36%), general (32%), priori (18%), typical (13%), expect (1%)
Class 3 : conditional (53%), frequency (35%), logical (9%), coherent (3%)
Class 4 : normal (49%), habit (39%), experience (12%)
Class 5 : none
Class 6 : mandatory.

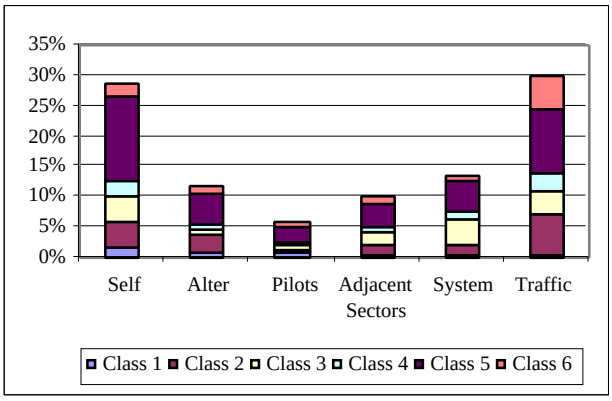


Figure 4 : Proportion of verbalization retained as a function of classes (defined in the terms-sample) and of different knowledge type (out of n=459).

If we leave aside class 5 (verbalization with no linguistic marker), the largest two classes are : class 2 and class 3 with correspond to knowledge (for example, we *know* that pilots call before x) and events that ought to occur (or frequent, logical or which are coherent). Class 4 corresponds to the wording of a default knowledge whereby the controller expresses reasoning out of habit or experience. This later class only accounts for 9% of observations.

The outcome of these results is that this knowledge not only concerns traffic but also, and by decreasing order of importance : one's own

activity, the behavior of the system, the teammate's activity, adjacent sectors and pilots. We will hereafter give a few examples that illustrate the way in which controllers rely on default knowledge.

- Knowledge about traffic :
A large part of this knowledge concerns the traffic. Typically, controllers make inferences on the descent profile of flights, their destination, their route or climb profile. Some examples we will give reveal that the deductions they induce, although they enable to diminish data acquisition, can in some instances lead to errors.
Example [1] gives the case of a flight (CFG) for which the controller does not check the flight destination (the destination is displayed). The controllers reasons on the airline and deduces the flight destination without cross checking via a data acquisition :
[1] EC : In my mind, the CFG goes to Luxembourg […] It is not a customary airline […] A bit like HLF earlier on. This proves two things : first when I integrated the CFG, I didn't look at its destination, it's a mistake from my part, and second: I don't have the information available on the table of flights.
In the following example, the controller builds a reasoning based on the exit beacon to deduce the flight destination. Note that this information (flight destination) is currently not available on the radar screen.
[2] EC : Yes there is a slight mistake from the start, I'm the one who screwed up. DITOR, for me, it's a guy (SWR106) who goes to DIJ whereas in fact it's a CTL. I don't interpret DITOR correctly, I only clear him up to 290.

- Knowledge about one's own activity :
This kind of knowledge can also relate to one's own activity. In the following example ([3]), the controller describes the way he uses a function (the alarm clock) because he considers (given the traffic) that his attention can decay.
[3] PC : When there is such a small traffic, an alarm clock is real good. That's what we are better off with. With an average traffic, when we keep going around the table of strips, when we really check virtually every single thing ; that's when we are efficient ; but when there's little traffic, your attention goes way down, and in that case, the wake up clock is not too bad, in those situations.

- Knowledge about the system :
This knowledge can also relate to the behavior one would expect from the system. In the following example, the controller comments on the system behavior, the creation of problem tags. He explains that in that case, he would not consult the label knowing that, given the traffic (transatlantic departures), he expects the system to create a Reminder label corresponding to those departures (with flight IT in climb). As seen before for Reminders labels consultation, the controller can in this case consider that he knows of the problem without consulting the label.
[4] PC : No, cause I knew that the IT climb, at that time of the day, with transatlantic departures, it is obvious that it goes in ; there is no point in wasting your time checking for the label.

- Knowledge about pilots and adjacent sectors :
Controllers have expectations as to their various counterparts : pilots and adjacent sectors. Regarding adjacent sectors, the controller can reason, depending on the context, on incoming calls ([5]), the adjacent sector's preferred conflict resolutions which will determine the way he will deliver flights to the neighboring sector ([6]) or else the implicit delivery conditions for some sectors, given their specifics ([7]).
[5] EC : I request something because that far away, you can't request him to give a heading because you don't know what the other guy has on another sector. Generally, he would have called to tell you there's a catch up.
[6] EC : what he refused, Barcelona, he wants them in parallel… Usually, they like it when routes are divergent because it rids them of all their extra traffic.
[7] EC : On the other hand, you must be aware that when you pass an aircraft to London, it is soon going to land like the BAW, we send him at 350, as soon as he is in contact, they consider it to be released. If we have folks bellow, we won't send him away, we'll only send him once crossed.
In the case of pilots, controllers integrate knowledge on the customary time of call from the pilot, or preferred habits (route constraints for fuel consumption, etc.). In the following example, the controller fails to recognize the incoming call due to a non customary time of call and the message content.
[8] EC : I'm so focused on my problem aircraft at Jersey that when there's an aircraft calling, which has nothing to do with it (AWD378) ; he wants JSY, it is a very seldom used fix, generally aircraft call before ORTAC. An aircraft which says JSY at first contact, it's really rare, JSY, bang, it's CRX I expect, I don't even think, I go "CRX106, proceed to JSY". She tells me 'that's not it but rather AWD378.

- Knowledge about teammate's activity :
The teammate's activity is also subject to this kind of knowledge. The planning controller can, for instance, make hypotheses on the kind of resolution that will be implemented by the radar controller. In example ([9]), he comments on the fact that his decision was based on the resolution he expects from his teammate.

[9] PC : The problem is that the radar controller, by culture, will be tempted, having once crossed this one (HLF) with that one (SWR), to put him direct on ABB rather than let him go there. We never do that, this fix on Monur. Direct ABB, he will cross here rather than there. I have to counter the action I know he will make, and therefore (x) behind. Find him a route that has him pass behind, direct CMB, it makes him turn right and so he goes behind.
[10] PC : It is no big trouble because it's obvious that he, at first contact, he will tell that ITHF to proceed GU. It's almost reflex like.

As we can see, controllers can make up inferences based on selective (or incomplete) data acquisition. We interpret this considering that the use of that kind of knowledge constitutes a cognitive trade-off between resources allocated to data acquisition and the building of a situation awareness. Such trade-off induces a risk in the sense that it can lead to errors ; however this risk taking can be limited by the enacting of representation updating. The idea for controllers is then to make decisions which will be updated pending the situation's evolution or pending one's own resources.
These results take part of a global understanding of how controllers used the system. We thus argue that observation on the use of the system make sense in the light of a cognitive model of controllers. The evaluation of the system could not go without the cognitive model which explains part of the system use. In particular, these results allow to evaluate whether the information display (requiring actions for data acquisition on the radar screen, or elsewhere, or no action at all) actually match the operational needs of controllers.

9. Conclusion

Data collected about the use of the ERATO tool showed different strategies in the way controllers use it. Further detailed analysis highlighted that these strategies revealed an economy of cognitive resources, tending to minimize data acquisition or input into the system.
A cognitive perspective has showed that default knowledge is a way to achieve this economy, and allows the controller to perform a cognitive trade-off. The principles and examples given illustrate how widespread the usage of this default knowledge can be and the fact that it can cover almost all aspects of the task. It also gives clues as to how some errors occur, which are the fallout of this default reasoning at work.

10. Discussion

One of the major conclusions of the presented results is to highlight the use of default knowledge by controllers seen as a cognitive trade-off between data acquisition (through the use of the system) and cognitive resource management.
Further results about default knowledge should be developed. A more complete analysis of the data, under progress, should examine cases where controllers claimed that the situation was not the one they had expected. Another challenge is to get a better understanding about data used for representation updating. Finally, we showed that controllers make assumptions about activities of other agents (teammate, adjacent sectors, and pilots). This point should be studied more precisely, to understand how the system helps or not for implicit cooperation between controllers.

Main global principles, for the design of the system, could be discussed:

- maintaining activities' variability in using tools : all controllers do not use their tools in the same way. For example, in the current environment, we know that the handling of paper strips may be different from one controller to another. Concerning the use of Erato, same data can be obtained in different ways. The possibility for controllers to carry out various actions would guarantee that they can adapt their method of work according to cooperation with the teammate, resource management or changes in the situation. Needless say how useful these results can prove when transferred into the design of tools capable of functioning correctly even when insufficiently fed with data inputs. Furthermore, the design of the HMI shall distinguish the utilization and the manipulation of a function and shall support the implicit cooperation modes between controllers.

- maintaining good situation awareness while controlling heavy traffic :
controllers should stay ahead of the traffic, which implies that relevant information is available to build mental representation of the traffic situation and to update it. Given that the ATC environment is characterized by a rich body of data evolving over time, their way of representing traffic situation and acting depends on expert knowledge about how the situation evolves over time : what we have called default knowledge (Leroux, 1993). The use of default knowledge can be seen as a cognitive trade-off contributing to resource management but seems associated with selective data acquisition. To keep good SA, controllers have to update their representation which refers to the process of checking that the actual understanding of the situation is accurate in terms of data (Abdesslem & al., 1999 ; Boudes, Bressolle, Leroux & Tremblay, 2000). This activity is strongly linked with cooperative mechanisms. Cooperation between the radar controller and the executive controller relies often on implicit communication (Bressolle, Pavard

& Leroux, 1995) which should be preserved (Bressolle & Leroux, 1997). Consequently, the new system should be an aiding tool related to these different activities, i.e. the use of default knowledge and representation updating which are part of controllers decision making process.

References

Abdesslem, S. ; Boudes, N. ; Bressolle, M-C. ; Capsié, C. ; Corredor, A-F. ; Leroux, M. ; Mesguen, L. ; Parise, R. ; Torrent Güell, M. & Tremblay, E. (1999) Evaluation d'ERATO 1997-1998, Rapport CENA/R98842/ICC, 517 p.

Amalberti, R. (1996) La conduite de systèmes à risques. *Le Travail Humain, PUF*. 242p.

Boudes, N. & Cellier, J-M. (2000). Anticipation accuracy in Air Traffic Control. *International Journal of Aviation Psychology*, 10(2), 207-225.

Boudes, N., Bressolle, M-C., Leroux, M. & Tremblay, E. (2000). Mental model updating process in air traffic control. *IEA 2000/HFES2000 Congress*, San Diego, USA, July 30-August 4.

Bressolle, M.C. & Leroux, M. (1997) Multimodality and Cooperation in Complex Work Environments : Some Issues for the Design and Validation of a Decision Support System in Air Traffic Control, In R.S. Jensen & L.A. Rakovan (Eds.), Proceedings of *the Ninth International Symposium on Aviation Psychology*, Columbus, USA, April 1997, pp. 212-218.

Bressolle, M.C., Pavard, B. & Leroux, M. (1995) The role of Multimodal Communication in Cooperation : The case of Air Traffic Control, In H. Bunt, R. Beun & T. Borghuis (Eds), *Multimodal Human-Computer Communication, Systems, Techniques and Experiments*, In Lecture Notes in Artificial Intelligence, Springer, vol. 1374, pp. 326-343.

Corredor, A.F. & Parise, R. (1999) Advanced decision aids for air traffic controllers : usability assesment from a cognitive point of view. Proceedings of *the International Symposium on Aviation Psychology*, Columbus, USA, May 1999.

Endsley, M.R. (1995). Situation awareness theory. *Human Factors*, 37, 32-64.

Endsley, M.R. & Kiris, E.O. (1995) The out-of-the-loop performance problem and level of control in automatio. *Human Factors*, 37, 381-394.

Leplat, J. (1996) Quelques aspects de la complexité en ergonomie. In F. Daniellou (Ed.) *L'ergonomie en quête de ses principes. Débats epistémologiques*. 57-76.

Leroux, M. (1993). The role of verification and validation in the design process of knowledge based components of air traffic control systems. In J.A. Wise, V.D. Hopkin, & P. Stager (Eds.), *Verification and validation of complex systems : human factors issues* (pp. 357-373). Berlin : Springer-Verlag, Nato Asi Series, vol. 110.

Reiter, R. (1978) On reasoning by default In R.J. Brachman & H.J. Levesque (Eds) *Readings in knowledge representation*, Morgan Kaufman Publishers, pp. 401-401.

Sarter, N.B. & Woods, D.D. (1994) Pilot interaction with cockpit automation : II. And experimental study pilot's model and awareness of the flight management system. *International Journal of Aviation Psychology*, 4, 1-28.

Woods, D.D. (1988). Coping with complexity : the psychology of human behavior in complex systems. In L.P. Goodstein, H.B. Andersen, S.E. Olsen (Eds.), *Tasks, errors and mental models* (pp. 128-148). London : Taylor & Francis.

Acknowledgements : Some results about default knowledge were obtained using a statistical tool called Stata ; the authors would like to thank Christine Maddalena for her participation at that level.

View publication stats