# Striving for Adequacy: The Importance of Rich HMI Requirements

**Alistair Jackson**
CORE Project manager
EUROCONTROL Experimental Centre
Brétigny-sur-Orge
France
alistair.jackson@eurocontrol.fr

**Alexandra Dorbes**
HMI Expert, CORE Project
EUROCONTROL Experimental Centre
Brétigny-sur-Orge
France
alexandra.dorbes@eurocontrol.fr

**Isabelle Pichancourt**
HMI Expert, C3T Project
EUROCONTROL Experimental Centre
Brétigny-sur-Orge
France
isabelle.pichancourt@eurocontrol.fr

## Abstract

This paper discusses some of the key issues, which have severely impeded the development of good Human Machine Interfaces (HMI) for air traffic controller working positions. It describes the approach to these issues that has evolved over a number a years within the EUROCONTROL Experimental Centre and emphasises the importance of HMI Requirements as a critical element in the design life cycle. A structure for clarifying and tracing requirements, based on constraints derived from these key issues is described. It is based on 5 levels, System Functional Requirements, System Mechanisms, Computational Requirements, Human Interaction Requirements and the Specification of HMI Mechanisms. The object is to allow separation of the effects due to the operational and technical concept and the interface design and realisation. It also provides a framework that clarifies the point at which different levels of design constraint should be applied. The use of this framework is illustrated with an example, drawn from system supported inter-centre flight data notification and coordination.

The paper concludes with a brief discussion on the use of the framework and issues associated with its potential for development.

## Introduction

This paper describes a structure employed within the CORE Requirements for Air Traffic Management Working Positions Project, at the EUROCONTROL Experimental Centre to capture some of the essential elements of Human Machine Interaction (HMI) requirements for Advanced ATM Systems. The structure is part of an evolving approach to HMI Specification, which began a number of years ago in the preparation of the Advanced Organisations for the first PHARE demonstrator (PD1). It has developed through our initial REFGHMI [1], the other PHARE demonstrators, [2] [3], EATCHIP III developments [4], and our association with the DSI Project [5]. It continues to benefit from observation of the development simulations conducted at the EEC on behalf of Member States. The current form is a response to pressures exerted by four seemingly very different development issues.

- How to achieve adequate specifications of the man-machine interface?
- How to address the classic problems associated with 'automation' and the division of activity between human and machine?
- How to establish adequate criteria for evaluation of human machine interaction?
- How to achieve the balance between harmonisation amongst our member states and the need to recognise and accommodate their different operational cultures and requirements?

Aspects of all four of these issues seem to converge at the level of definition of requirements for human machine interaction.

## Elaboration of the Issues

### HMI Specifications

In recent years controller working position HMI has often been seen to be at the centre of difficulties in attempting to upgrade ATM systems. Many factors contribute to this situation. For example:

*HMI as an Integration Focus:* In even the most modular system development process, a majority of elements have to be in place before the HMI can be animated and fully tested. Ironically, until the HMI is in place many elements of the system cannot be functionally tested in an 'operational' context. A lot of integration problems only become visible when the HMI is under test.

*HMI creates the System Image:* The HMI is the part of the system visible to users. The presentation and behaviour of the HMI shapes users' understanding of the system and their perception of its performance. Users cannot easily separate HMI and other parts of the system. Poor HMI produces the impression of poor system functionality and poor system functionality is often blamed on the HMI. Further, the HMI is generally the end user's first real exposure to the developing system and to any problems associated with integration.

*The Reality gap between Customers and Suppliers:* There are three important facts to note.

1. In ATM (unlike the aircraft industry) the high level design of the system (and the HMI) is very frequently performed by the ATM (service provider) clients and not by the supplying industry.
2. Software produced in development laboratories, or for HMI in ATM simulation, does not need to con-

form to the same standards of robustness or completeness as are necessary for industrial implementation of ATM systems. This is also true of the data used to animate controller support tools.

3. HMIs are judged as a whole or with individual symptoms of failure being reported. It is not always clear from ATM simulations and experiments, which individual elements of any particular HMI are solid and robust, and which are barely acceptable.

When these facts are taken in conjunction it becomes clear that there is a risk that a client will specify ATM HMI functionality which cannot be realised in practice. This has already happened on several occasions.

Often, inadequacies of HMI specification are blamed for difficulties encountered at the HMI testing stage but arising from integration problems and inadequate system functionality. There is a need to be able to clearly identify the relationship between general system functionality and HMI functionality so that testing and revision can be focussed and responsibility for remedial action correctly allocated.

*HMI Software Explosion:* There seems to be a clear consensus that graphical user interfaces based on windows and direct object manipulation are the way ahead for controller interfaces. However, this implies a very substantial increase in the amount of code necessary just to manage the controller interaction. This is coupled to the fact that we are also seeking to integrate many more sources of data into the control environment. Our 'naive' estimate is that there is 8 to 9 times more code in the HMI of next generation systems as was required for the entire ATC system previously. All of this HMI functionality needs to be specified, tested and managed in detail.

Completeness of the detailed low-level specification required for code realisation is often only possible for systems '*which you already know how to build'*. The risk of incompleteness is high in new generation HMIs.

*Consequences for the Format and Use of 'HMI Specifications':* As specifications become larger and more detailed, difficulties arise with the structure and management of HMI specifications themselves. Documents bearing this title exist at many levels of description, from high level HMI requirements (e.g. 'the controller will be able to update cleared flight level') to the detailed formalisms used to guide software engineers in code production. This is a reflection of the fact that HMI specifications serve a number of roles. Successful production of specifications is a complex process involving the integration of several different types of expertise derived from the interaction of a number of different actors. In a previous paper [6], our group has attempted to describe this context more fully.

The risk of incompleteness implies that the rationale behind design decisions should be traceable right through the specification process. This allows remedial actions taken late in the development process to merge consistently with the rest of the design.

Our current specifications involve global principles, textual descriptions of functionality, textual formalism for dialogues, qualifying descriptive text, illustrations of graphical presentation, and state transition descriptions. In reality they remain inadequate in many ways and suffer from important difficulties of traceability, navigation and management. Given that the software community has recently made significant advances in this area with the widespread use of the Unified Modelling Language and the growth of support environments for document control, software management, etc., we are now attempting to integrate both our approach and materials into these environments. The activity seems most promising and the objective of realising end to end traceability may be practicably attainable.

**Conclusion:** *There is a need to be able to produce detailed and complete HMI specifications. At the same time there is a need to be able to trace the design rationale throughout the specification process to support consistency and remedial actions in the case of incomplete specification.*

**Automation: How to Avoid the Issue**

In the early '70s a number of key papers sensitised us to the issues associated with the introduction of 'automation' into industrial and bureaucratic environments [7], [8]. Since that time debate on the automation of controller functions has occupied the ATM community to an almost obsessional level. The initial debate focussed on the allocation of tasks to human or machine based on the notion of their relative strengths and weaknesses (following Fitts [9]). The debate continued through the 80's. On the one hand were those concerned with '*the elimination of human error*' and '*removal of the controller workload bottleneck'*. On the other were those who wanted to keep the controller '*in the loop'* and saw the human as the *'last line of defence'* when the unexpected occurred.

The focus shifted to the consideration of different types of relationships between human and machine (e.g. the 'electronic cocoon', adaptive decision aids of Rouse [10] and the 'nine automation modes' of RHEA [11]).

While this approach provided many warnings of the perils of automation, with a few exceptions, it provided little insight on how to proceed. Negative results were readily attributed *post hoc* to automation but the approach was generally unable to predict which of the particular undesirable characteristics 'classically' associated with automation, was likely to apply in any particular design context. Further it was very difficult to translate the admonition *to keep the man in the loop* into design practice or even into evaluation criteria.

Heinz Erzberger (responsible for the development of the CTAS tool) inspired an alternative approach. The Erzberger 'Guidelines' [12] provided a list of Do's and Don'ts to be applied to the inclusion of automation in ATM. In fact, these criteria were a set of judgmental 'craft rules' to be applied by designers. This approach, of *designing to avoid the negative consequences of automation*, has been extended in our own HMI work; firstly, as a set of additional design principles ([1], Chapter 6), but more recently in the form of a design strategy.

Effectively, we have taken the position that most (not all) of the negative consequences associated with automation are not unique to automation. They can be associated with any type of change in the work environment irrespective of the origin of that change, e.g. organisational re-structuring.

As an example, if we consider Endsley's well known classification of relationships between human and computer [13], and we then substitute a second person for the computer; what is changed? We would argue that the effects are similar and that it is these effects and consequences, which have to be managed and become the measures of our design.

Norman [14] had already pointed firmly in this direction by showing that some important 'automation effects' were in fact feedback effects. Quality of feedback is something a designer can work with. By analogy, issues like loss of transparency, autonomy are not to be seen as potential negative effects of automation but as characteristics of the ways in which jobs and tasks are realised. They should be turned 'inside-out' to produce criteria for good design.

Once a human agent has been identified in the system, the designer's task now includes the need to assure that the job of the human is a 'good' one. In our more pompous moments we call this Pro-Active Job Design.

In our experience, if a job is simultaneously:

- interesting (motivating),
- has clear responsibilities which can be directly addressed and controlled by the job-holder and
- gives good clear feedback as to results, encourages the development and demonstration of skills (supporting good performance and satisfaction),

then the job-holder is not concerned as to whether a computer is present or not.

Basically, we address automation by changing the question and directly addressing the issue of designing a job that is not subject to the 'classical' negative characteristics associated with automation. This need for explicit job design generates constraints, or additional requirements for the design process. We need a framework in which they can be specifically included.

**Conclusion:** *The requirements capture process must provide a level of description which can specifically address the reconciliation of human requirements and those characteristics which are commonly regarded as the negative aspects of automation. Decisions on the allocation of responsibility, (ownership of information and actions) closure of tasks and initiative should be clearly visible and justified.*

### Issues of Evaluation and Context: Striving for Adequacy[1].

Testing has to take place at many different levels of system development. This is especially true if we want to separate out the effects of the HMI design, implementation and performance from those related to other aspects of system functionality. There is no other way to separate the effects of HMI from the impact of the system concept and to avoid the confusions described in the section on HMI specifications.

This seldom happens. It is a harsh truth that, as a community, our current system evaluation processes are inadequate. Despite many serious attempts to improve this situation, many of our large-scale pre-operational evaluations are much closer to demonstrations of feasibility[2] than measured activities. When something works (or more particularly does not work) we are often incapable of separating out the different effects that have contributed to the result. The impact of a new concept, cannot be readily be separated from effects of HMI design, HMI implementation, user training or user performance. How do we deal with this situation? Our only current suggestion is based on analogy with the models employed by the software community in terms of decomposition and incremental testing.

For each level of contributing effect, criteria must be established on which the testing can be based. In a previous paper [6] we described an extended version of the Classical V model of system development (Figure 1). The V model was chosen because, in spite other limitations, it emphasises the evaluation component, and the need to define traceable objectives, criteria and test plans throughout the system design process. In particular we introduced the distinction between Functional Requirements and HMI Requirements. The latter include additional requirements that come about as a result of design decisions made in terms of the roles of the human participants and the resulting job designs.

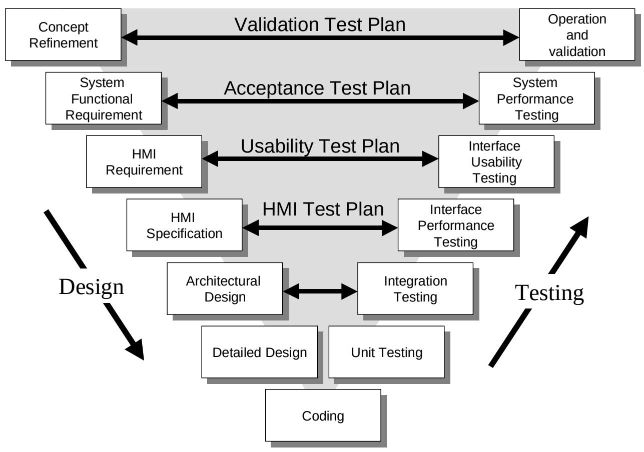


Figure 1: Elaborated development cycle emphasising testing

For the previous paper, our principal concern was with better understanding the requirements capture processes. However, the natural symmetry of the 'V' led us to have

[1] The 'Striving for adequacy' motto began as a reaction to counter-balance statements made about 'optimal HMIs'. In fact adequacy in a demanding domain like ATM is a very high standard indeed.

[2] This is not to say that they are not of considerable value and necessary steps in our validation process. They could just be better.

a better understanding of how to decompose testing to evaluate HMI aspects[1], prior so the global human system and operational concept levels. The natural division between HMI Requirements and HMI Specifications also seemed to map naturally onto a separation between broader usability testing and testing of HMI performance.

One of the commonest misconceptions about design is that there exist perfect solutions that will apply under all circumstances. In reality, design (HMI design in particular) is heavily context dependent. All design decisions are trade-offs; every advantage gained has a cost elsewhere. A good HMI is one in which the trade-offs reflect the priorities of functions, users and context for which it is designed. For example, operational validation can take place only 'in the operational context'.

In order to make trade-offs correctly it is necessary to understand the requirements. In order to test whether trade-offs have been made correctly it is necessary to have clear evaluation criteria. Explicit evaluation criteria can only be based on clearly stated requirements. The evaluation criteria should be established at the time the requirements are identified. Test plans should be established as the design progresses and both the requirements and the assumptions behind the trade-offs being made should be traceable through to the eventual testing activity.

**Conclusion:** *Several different levels of testing are only possible once the controllers and the HMI come together. There is a need to disentangle what is attributable to the HMI from what is attributable to other aspects of the system. This implies the need for clearly established requirements, a clear separation of functional and HMI requirements and traceability of these through to final testing.*

Another important aspect of context, is that very little of our system design today takes place in a holistic top down way. Most of the time we are looking at the creation of sub-systems within the constraints, procedures, or manning practices of existing systems. As well as understanding the consequences of change in system design, it is also very useful to be able to understand the consequences and rationale associated with the constraints we retain. This not only helps us to understand the context for any current project but to recognise other areas that may benefit from change.

**Conclusion:** *The Requirements Capture process should also be able to make explicit the nature and consequences of those constraints inherited from the existing system or arising from other design decisions.*

### The Need for a Generic Approach and the Role of HMI Requirements

The service role of EUROCONTROL raises special questions for HMI designers. As an organisation our role is to support a wide range of member states and yet in the previous section we noted how HMI design is heavily context dependent. One of the main reasons that our group initially began to place so much emphasis on the clear statement of HMI requirements to support particular ATM functions, was the belief that this was a practical approach to HMI harmonisation. We contrasted this positively with the idea of HMI standardisation at the level of HMI mechanism, look or feel. We felt there were three reasons for the advantages:

1. As already noted, most new ATM functions are added into the context of existing systems with established constraints, including HMI look and feel[2]. Importing a function with a fixed look and feel would almost certainly
   - create inconsistency with the rest of the HMI
   - require re-engineering of the imported component, or
   - require modification of the look and feel of the rest of the system HMI

   None of the above are considered as good options.
2. Taking ownership of a new system or function is important for the users. The HMI is the most obvious manifestation of the system or function. Few groups are willing to accept a 'second hand' HMI. Some customisation is almost always necessary and, if intelligently conducted, the process of customisation can serve to help users 'take ownership' of the changes.
3. Local context will change the balance of trade-offs. Needs will be different.

This reasoning led us to the conclusion that the most general and practical HMI product for a new function was a clear statement of the requirements that an HMI would need to meet in order to support that new function. This should be complemented with illustrations of how different exemplary HMI mechanisms would meet these requirements with different balances of trade-off. The potential customers could then identify the types of solution most relevant to their environment. This is what we refer to as the 'Rich HMI Specification'.

Systems meeting these common HMI requirements would be potentially inter-operable a functional level.

***Conclusion:*** *The 'Rich HMI Requirement' could provide an effective method for providing harmonised solutions in a variety of related, but varied, contexts. It would allow the intelligent adaptation of the description, systematically to different contexts.*

## The Current CORE Framework

This section describes an evolving framework for elaborating requirements, which tries to address some of the concerns above. It seeks to provide:

a) a framework in which the nature of design decisions is more transparent and traceable;

[1] Within the CORE project, the CORE Platform is specifically conceived to support the HMI evaluation steps within a technical context identical to that which will be used later for simulation and concept evaluation. This is necessary to allow end-to-end traceability

[2] In this context we consider 'look and feel' to be set of metaphors, conventions and mechanisms which suggest how a particular HMI requirement can be realised ***consistently*** with the rest of the interface.

b) identifiable points at which specific classes of design issue can be addressed;

c) a means of making certain key aspects of context more visible and allowing potential users to see at what point things must be changed to modify the description to better support their particular context;

d) a means of reinforcing the connection between requirements and evaluation and allowing traceability of the origin and nature of requirements to allow separation of their effects during the evaluation processes.

e) a decomposition that makes the process more manageable without losing the structure and relationships which integrate the process (decomposition without fragmentation).

This particular version of the framework has evolved in the context of trying to establish an understanding of a full implementation of the SYstem Supported Coordination (SYSCO) extensions to the On-Line Data Interchange (OLDI) standard [15]. We are currently seeking to establish its generality as a tool in thinking about any system design context.

**The Framework: a Ladder of Requirements**

The framework consists of five steps in elaborating a System Functional requirement down to a detailed HMI specification (Figure 2). Each step is represented by another level of description. In moving from one step to the next, different classes of contextual constraints are brought to bear, different design issues come into focus and different questions can be addressed. Since, these appear to be stable across different levels of detail of functional requirements it apparently becomes much easier to identify where certain design issues are located. This supports improvement of completeness and traceability.

The five steps of description are:

1. System functional requirement
2. System mechanism
3. Cognitive/computational impact and requirement
4. Minimum Human Machine Interaction Functional requirement
5. Human Machine Interaction presentation and mechanisms

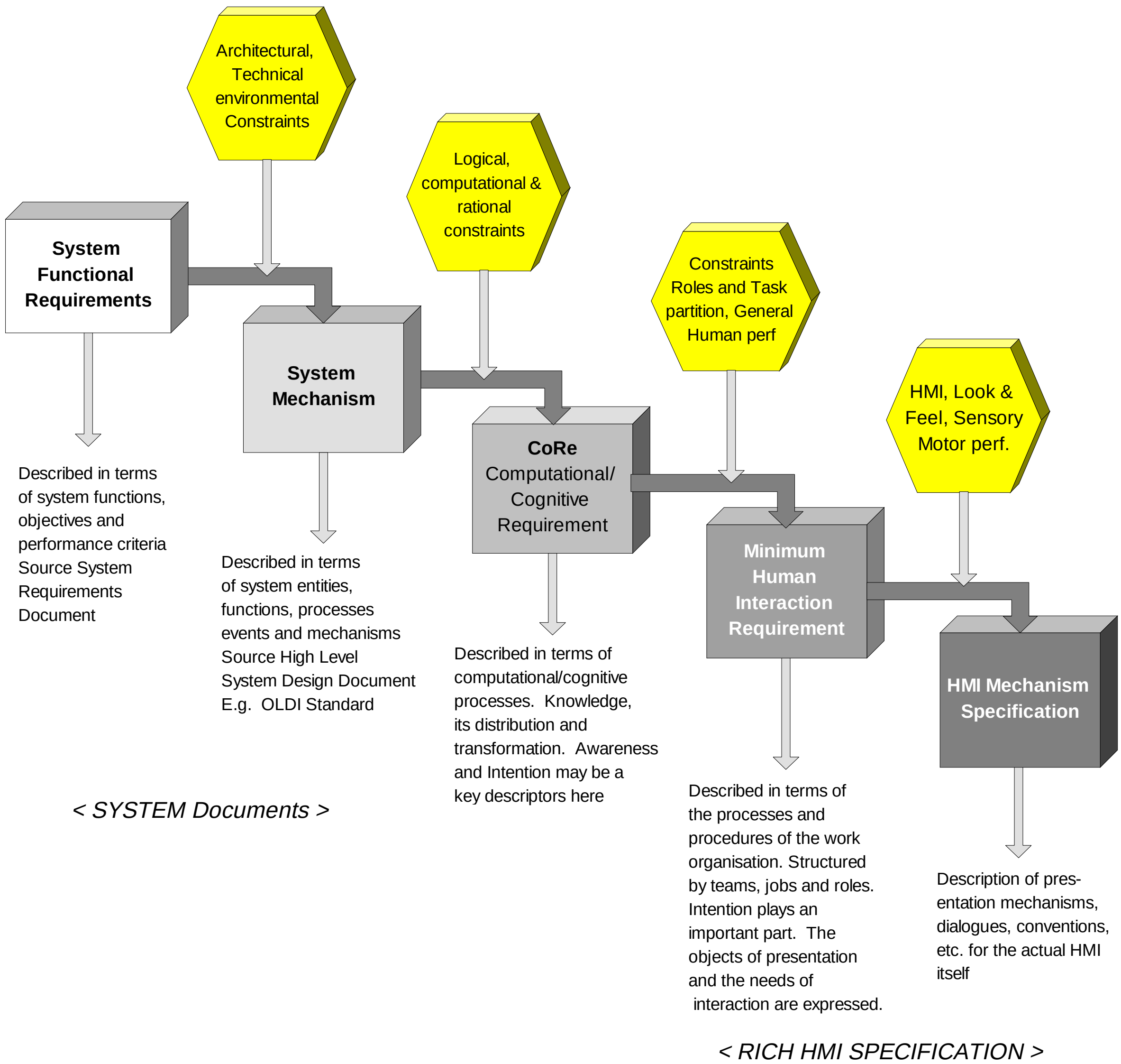


**Figure 2: The 5 Steps in the Requirements Framework**

The first, second and fifth of these descriptions are comparatively familiar. The fourth, HMI Requirements, is an idea we have been seeking to elaborate for the last couple of years, as part of our attempts to establish a generic approach. The innovation lies with the third step, currently labelled, somewhat clumsily as the 'Cognitive/computational Impact and Requirement' (CoRe) for short. It was 'forced' on us by the realisation that a number of factors were confounded in our original perception of HMI requirements.

We shall describe these five steps, supporting them by consideration of an example based on the use of the Notification Messages in the OLDI/SYSCO standard. The example is based on simplified requirements but we hope that it serves to illustrate both the nature and the generality of the approach.

Note that the requirements increase in specificity as the different steps are traversed but they also change qualitatively as new constraints and factors are brought into the design process.

Figure 2 illustrates two additional aspects of the framework. The lower part of the figure associates each of the steps with a corresponding vocabulary of objects and processes. The upper part of the figure shows different types of constraints which have to be considered in the development of the system and the points in the process by which they should have been injected. For the sake of graphical clarity these are shown as being injected into the transitions between states to indicate that they already function as constraining influences to be applied to the following step. In fact it is the '*taking-into-account'* of these constraints, which drives the transition to the following step.

In consequence, the following description will define each of the five steps in terms of:

- its function,
- the (typical) sources of the contributing information,
- and the language of description
- the constraints which may be associated with it,

Where possible, we make some reference to identification of performance criteria.

**Step 1: System Functional Requirements**

This is a high/medium level description of the System Functional Requirement. It will define the scope and context of a subject system or function[1] in terms of its relation and dependencies with other sub systems and functions.

It will typically be based on the operational requirement documentation.

Performance requirements will be described in terms of contribution to overall (external/global) system performance and system level objectives.

***Example (OLDI/SYSCO[2]):*** *Basic Flight plan data is made available to the controller and the unit software system prior to aircraft arrival.*

*This can be*

- *The availability of a new flightplan*
- *An update of a pre-filed flightplan*

*Performance is defined in terms of the reduction of human support activity and improved efficiency*

**Step 2: System Mechanism**

This step describes the mechanism employed by the subject system to meet the requirement expressed in Step 1. It is expressed at a level of detail, which is more specific than the System Functional Requirement to which it corresponds. It will reflect the high level technical design; the choice of technical solution chosen.

Typical source documentation would include the high-level system design documentation.

The description will include reference to architectural elements of the system already described at this level. It will be in terms of system entities and functions, processes, events and mechanisms.

It will typically explore the effect of external technical constraints, communications, protocols, etc with external entities, data and time constraints. Interfaces to other external, functions in the system will be referenced in more detail.

Performance criteria will be expressed in terms of the objectives of the system/function and of the performance and efficiency of the protocols, process and mechanisms in enabling the subject function to perform as required.

***Example:*** *For the example the system mechanism is provided by the OLDI messaging system and is described in terms of the transmission, reception and timeouts of the OLDI standard. More specifically it relates to the use of the NOTIFICATION functions supported by a set of messages with defined structures and protocols:*

| | |
|---|---|
| *Advanced Boundary Information* | *ABI* |
| *ACTivation message* | *ACT* |
| *Referred Activation Proposal message* | *RAP* |
| *Preliminary ACtivation message* | *PAC* |
| *Message of Abrogation of Control* | *MAC* |
| *Logical Acknowledgement Message* | *LAM* |
| *Stand-by Message* | *SBY* |

*These message pass between entities identified as 'transferring' and' accepting' units.*

*Performance measures could be described in terms of transmission and reception rates, channel availability, and transmission delays.*

[1] At this level of description 'function' and '(sub-) system' are used almost interchangeable to describe the object of a particular development and evaluation activity (*the subject system* in our text). They are juxtaposed to the 'global system' which provides the context in which they contribute. The use of the two terms suggests the scalability of the framework we are proposing.

[2] Full OLDI/SYSCO requirements can be found in [15].

**Step 3: Cognitive/Computational Impact and Requirement**

As its name suggests this step attempts to describe the key information processing activities which must take place in the subject system/function if it is to achieve its objectives. The source is not explicitly documented except through the previous steps. The input of domain experts is key to this stage. Within the data constraints identified in Step 2, the information needed to perform the function is identified together with the representations and transformations necessary to produce the desired outcomes.

This is the *cognitive impact* of the function.

In many cases this is mainly the implication for human processing but, if other agents such as trajectory prediction tools were to be considered, the discussion could equally include them. Description is typically in terms of different knowledge elements, transformations of knowledge and cognitive functions of the system, e.g. identifying, associating, comparing, indicating, integrating, separating, etc. The system agents conducting the processes will be differentiated but only in broad terms, e.g. supervisory function on accepting unit. Processes will then be clustered (on the basis of minimising coupling) into roles. Normally, roles are not fully allocated to agents until the following step but if certain constraints such as the existence of certain system functions (e.g. trajectory predictor) or of manning practice (predefined planning and tactical roles), assignment to specific agents may already be known.

Performance is in terms of the availability of correct information and the success of the various cognitive processes within the known time constraints. Completeness and lack of ambiguity may also be key elements here. Much of the analysis of possible system error states takes place at this step and the following one. Similarly issues associated with automation and situational awareness begin to be addressed here.

This is the step at which the logical feasibility or not of the function is really evaluated.

***Example:*** *Discussion is in terms of the information available (the knowledge states) and its consequences in the two identified units (transferring and accepting). For example, a Logical Acknowledgement Message (LAM) is normally*[1] *sent (by the accepting unit to the transferring unit) in response to an Advanced Boundary Information message (ABI). If the LAM is not received on the transferring unit there is a state of ambiguity. The transferring unit does not KNOW whether the accepting unit has failed to receive the information (ABI failure), or if the acknowledgement (LAM) has failed. These two conditions may have very different consequences but both of them require a resolution initiative from the transferring unit. There is a third condition where the ABI is received and the information it contains correlates but with discrepancies against the pre-filed information on the flight. In this case the LAM is not sent and resolution initiative lies with the accepting unit.*

*In this case, performance could be expressed in terms of the complexity of the situations which results, the non-nominal conditions which can arise and how completely they can be addressed.*

**Step 4: Minimum HMI Requirement**

This description translates the consequences of the previous steps into requirements for the design of the Human Machine Interaction. These fall into two categories; the *Primary HMI Requirements* derived directly from the system requirements and *Secondary HMI requirements* that derive from design decisions which are made about the nature of the human roles and general requirements for optimising human performance. To achieve this tasks and functions will be clustered (on the basis of de-coupling) into roles and assigned to agents[2]. The choice of agents may already be constrained by context, defined and inherited from previous steps, i.e. a function being introduced into an existing manning configuration, but equally the need for new agents may be identified. The same processes will establish the separation between human and system agents.

The detailed assignment of different roles to individual agents and the communications requirements between the agents is one of the important sources of secondary HMI/human factors requirements. Job design takes place at this level, balancing of workload, individual and team-resource management, establishing feedback and closure; in general dealing with those issues identified in the section on automation.

The resulting description is in terms of:

- the processes and procedures of the work organisation, structured by teams , jobs and roles
- The repertoire of intentions and the vocabulary of operational actions for each of the agents in different contexts, together with the information and data on which these operations take place
- requirements for the prioritisation and performance of these actions and a description of the range of contexts in which they can occur.

Associated performance criteria relate most closely to what is frequently identified as usability and fitness for purpose. Criteria derived from the primary HMI requirements will dominate in dealing with purpose, i.e. how the well the user (s) and resulting HMI will be a able to meet the system objectives. The traceability of the requirements plays a key role here. Both primary

[1] If the ABI information correlates correctly with pre-filed information for the flight or if there is no pre-existing information and a new flight plan is generated.

[2] One agent may assume more than one role. A role is a group of logically and intentionally related tasks, e.g. Flow management, coordination. Often a 'large' role will be identified with a unique position/agent a job e.g. Planning Position/Controller

and secondary requirements will contribute too criteria for addressing the 'fitness' aspects. Here we will find issues as diverse as the coherence and quality of the jobs and feedback to the effectiveness of the team communication

In summary this step describes in detail what the HMI and staff collective must be able to do and allow in different circumstances. It is the basis for the HMI specification in the following step but also for the detailed operational procedures and working practices that are supported through it.

***Example:*** *In moving into this step we are faced with introduction of constraints relating to the roles and agents involved. In the case of SYSCO this will depend on the context of manning practices. For example a core European implementation within an EATCHIP III type scenario we might have to assume:*

- *a Planner/Tactical controller (PC/TC) pairing with PC responsible for co-ordination of entry and exit conditions,*
- *flexible task assignment between PC and TC.*
- *a Centre Supervisory role*

*And we would have to make a decision on:*

- *Whether, within an ATC unit, inter-sector coordination works in the same way as inter-unit coordination? (This makes for consistency but places limitations on inter sector coordination.)*
- *Who is responsible for the monitoring of the integrity of flight data at a unit level? Is it the supervisor or is there an additional role, e.g. could it be linked to a flow management or multi-sector planner role?*

*The resulting description indicates what information and interaction options should be available to each role and in what context. It should describe working procedures and exception conditions. In our example, if the ABI is sent and successfully processed and the corresponding LAM is sent and received then:*

- *on the accepting unit the HMI would need to indicate the availability flight plan data to the TC and PC and provide them with a means to access it as appropriate.*
- *on the transferring unit – no indication is necessary*

*In the set of conditions where the LAM is not received to the ABI, we could imagine:*

- *the need for an alternative communication mechanism between the transferring and accepting units (e.g. telephone)*
- *for each of the non-nominal cases, the presentation of the necessary information with an identified priority to appropriate agents (in the ABI case a variety of solutions are possible but the TC would <u>not</u> normally be involved)*
- *for each case a means of updating the system information following resolution*

*Performance criteria at this level typically relate to the usability of the resulting HMI. In our example the clarity of the procedures; the ease with which the condition, and the responsibility, is identified; the disruptiveness of the condition could all be the basis of metrics. Note that the frequency with which the non-nominal situation arises is more likely to be a metric of the 2$^{nd}$ or 3$^{rd}$ steps as opposed to the HMI requirements. This is a good example of the type of failure, which could otherwise be incorrectly attributed, by users to the HMI. It reflects the usability of the system NOT of the HMI.*

**Step 5: HMI Mechanism Specification**

The final step in this requirements process is the translation of these HMI requirements into a set of HMI specifications, which will be used to produce and test a suitable HMI. The specification consists of interaction mechanisms, dialogue structures and HMI performance criteria. This product is comparatively familiar and corresponds to the HMI description provided to software engineers as the basis for the software design of the HMI.

The actual process followed in its production will depend on circumstances. When adding a new function within a well-constructed and documented system, little innovation may be required. It is imaginable that, if there is already an adequate system style guide dealing with look and feel, basic interaction mechanisms, visual conventions, metaphors, etc., then mapping of the requirement onto the guide may generate the specification almost automatically.

In the absence of such 'HMI principles documentation' or to the extent that the new function has new requirements, innovation in design will be required. In any case the main sources of information are the HMI requirements from the previous step plus Basic HMI principles which act as constraints. The HMI principles are general, relating to the type of interface medium selected[1] and more specific psychophysical elements relating to the specific types of interaction mechanisms under consideration.

The language of description will be in terms of interaction primitives, objects dialogue structures etc.

Typical evaluation criteria, will be response speed and error probability of both system and human action. Consistency, legibility, physical and perceptual discriminability will be typical dimensions of evaluation.

***Example:*** *This will depend on the look and feel being employed. If we assume the format and HMI conventions currently applied in CORE, which is very similar to that of the DSI prototype [5], then the set of mechanisms would be:*

***For the case of successful ABI/LAM processing***

[1] In most cases today this will be a direct manipulation graphical interface.

- *on the accepting unit, following transmission of the LAM, the aircraft radar label presentation on both the PC & TC radar displays would change from the 'uncorrelated state' (a/c symbol + squawk + mode c, no label interaction) to a correlated state' (squawk becomes callsign and the information mouse button functions allowing display, but not manipulation, of additional flight data become available).*
- *no indication on transferring unit*

The next part of our description is couched in terms more suitable to the HMI requirements level (rather than the HMI mechanisms level) because we are not citing a specific system and context, which would normally be the case

***For the case of correlation with discrepant data***

*on the accepting unit,*

- *indication of the discrepancy is shown to the role associated with Flight plan integrity. (This could be first PC, Supervisor function or specialist position). This agent phones the transferring sector and updates information through a purpose defined HMI.*

*on the transferring unit,*

- *LAM failure to ABI is shown to the Supervisory position (documented OLDI requirement) as it may be indicative of a major communication failure. A call may be received from accepting unit.*

***For a LAM timeout on the transferring unit,***

- *on the accepting unit, where the aircraft may, or may not have received the ABI, there may be no indication visible.*
- *on the transferring unit the LAM failure to ABI is shown to the Supervisory position (documented OLDI requirement) as it may be indicative of a major communication failure. A responsible agent needs to establish contact with the accepting unit and restore data integrity on the two sides. We shall not explore all the necessary steps and actions.*

*Performance criteria will differ for the different mechanisms in question. For example in the case of the successful ABI/LAM, the criteria may be that the update DOES NOT attract the attention of the TC or PC when involved in another task – background update. On the other hand the discrepancy conditions may need to attract the attention of the appropriate agents within a certain time threshold. It should be noted that given the role of the ABI, typically 30 minutes before aircraft transfer this is not normally a time critical event.*

## Observations and Issues

Our experience with using this framework is still in its early stages. Over the next 18 months it will be tested and evolved through its use in the CORE project. Both as a tool in improving our capture and description of requirements and as a support to our HMI evaluation activities. To date we have collected a series of observations about its use and a set of issues to be resolved.

**Using the Framework**

From even the limited example cited above, it is clear that application of this process is not simple. The process of collecting requirements is a fundamentally difficult one. Reference [16] illustrates the complexity of a well-produced requirement document in the context of the Short-Term Conflict Alert Function for ATM Safety Nets. In that example we find requirements mixing all the categories implied in our classification.

Currently, we are still exploring the use of the framework. Our present technique is an extension of previous HMI practice and is based on performing a 'scenario walkthrough' of the typical applications, e.g. an aircraft transiting the airspace we are considering. In the course of this walkthrough we consider the behaviour of different system elements as significant system events arise. The exercise will often begin with a simplest case example and then be elaborated to deal with complex and exception cases. This typically results in a matrix of the type shown in Figure 3 where require-

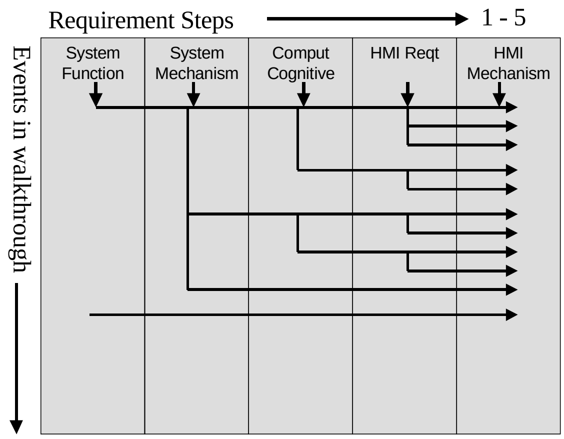


Figure 3: Event Walkthrough Requirements Matrix

ments (represented by the arrows) increase in number and specificity left-to-right.

The technique is useful, supports some aspects of traceability well, but the nature of the design decisions and the effect of the constraints introduced at each step is not yet sufficiently clear. The analysis is time consuming and extensive but has the advantage that, once the outline functional requirements have been established the rest of the table can be completed in consultation with different experts. The resulting material forms a good basis for considering the effects of changes. This is promising when considering the potential for re-use.

**The Importance of Non-Nominal Conditions**

From the ABI example above it is clear that a great deal of complexity can arise from the non-nominal conditions and their management. We have found this to be generally true. We have also found that when a great deal of complexity suddenly arises at any one step it can sometimes be reduced by a re-appraisal of requirements in the previous step[1]. Current system 'validation' activities based around large-scale real time simulation tend to avoid these non-nominal situations but ironically, to win operator confidence, it is just these conditions which must be clearly manageable. We foresee increasing emphasis on these aspects and in this context will seek to develop a closer cooperation with the HERA project [17].

**The value of the Computational and HMI requirements Steps (3&4)**

Steps 3 & 4 are the focus of the key human Factors issues. We are increasingly convinced of the value of the separation between the cognitive and HMI requirement steps. For example, we identified a case in SYSCO where it had been assumed that refusal of the Referred Activation proposal message would allow the system to revert to a previous state. We identified that at the cognitive level this was not true. The controller's now had awareness of a potential problem and this could have an effect on workload and monitoring strategy. This separation between changes in knowledge state and how these changes are managed in the procedures and HMI seems essential for good job design and in ensuring coherence between the system image created by the HMI and the real system state (good situational awareness?).

**Starting Point and Completeness**

In most of our discussion the progress through the steps has been described as left to right, but this is an ideal situation. In the most common case, where we are making a change within an existing context, the change could begin anywhere. For example, a re-allocation of controller tasks would probably begin at the HMI requirement (step 4). A concept like 'Shared Pilot/Controller Information & Awareness' [18] has its origins at the Computational/Cognitive level (step 3). In fact, by the time they are ready to be subjected to detailed analysis, most ideas for change will reflect requirement and justification at several levels. In the case of pilot/controller information sharing, the existence of the datalink (a system mechanism, level 2) is a necessary enabler and may even have been the original driver of the idea – as a kind of technology push.

What is important is that the effect of changes is pursued across the requirement steps to manage its impact and to establish the criteria, which will isolate and assess its contribution.

In addition to the cascade of requirements, change at any level may block the realisation of requirements from the previous level. Requirements often have tolerance[2] and can be reformulated to accommodate the consequences of the change.

One way to visualise this is that *requirements* propagate left-to-right from steps 1 to 5, *constraints* on the other hand propagate in the opposite sense. A kind of negotiation back and forth, balancing the possible against the desirable, may be necessary to reach a workable solution.

## Summary and Conclusions

In the context of the multi-disciplinary development of control working positions we have described a framework which is intended to help manage the identification of human machine interaction requirements. This structure tries to segment requirements in a way that clarifies the stage of development at which different classes of constraint and design issue should be incorporated and addressed. The key aspect of the approach is improved traceability of different elements of the requirement throughout development. Traceability has two specific objectives: the development of adequate and appropriate test criteria; and more efficient re-use of requirements through better understanding of the scope of changes. As part of the CORE project, the framework will be subject to evaluation and revision to establish its usefulness and generalisability.

## Abbreviations and Acronyms

| | |
|---|---|
| ABI | Advance Boundary Information |
| ACT | ACTivation message |
| ATM | Air Traffic Management |
| CTAS | Center TRACON Automation Systems |
| CORE | CORE requirements for ATM Working Positions, a project managed within the EATMP Human Resources Sub-Programme |
| DSI | Denmark Sweden Interface |
| EATCHIP | European Air Traffic Control Harmonisation and Implementation Programme |
| EATMP | European Air Traffic Management Programme Successor to EATCHIP |
| HERA | Human Error in ATM Project managed within the EATMP Human Resources Sub-Programme |
| HMI | Human Machine Interaction/Interface |
| LAM | Logical Acknowledgement Message |
| MAC | Message of Abrogation of Control |
| OLDI | On-Line Data Interchange |
| PC | Planning Controller (organique) |
| PD1, 2, 3 | PHARE Demonstrators 1, 2 &3 |
| PHARE | Programme for the Harmonisation of Air traffic management Research in |

[1] We have previously observed a related phenomenon at the level of HMI mechanism design where highly elaborate mechanisms frequently reflected inadequate requirements

[2] Often it is easier to 'overspecify', in the sense of demanding unnecessarily high performance, than to perform the analysis for an accurate specification. This is one of the many reasons for the expectations gap.

| | |
|---|---|
| | EUROCONTROL |
| RAP | Referred Activation Proposal |
| REFGHMI | REFerence Ground Human Machine Interface, a synthesis of ODID and PHARE PD1 basic HMI functions. |
| SBY | Stand-BY message |
| SYSCO | SYstem Supported COordination |
| TC | Tactical Controller (executive or radariste) |

## References


[1] **Jackson, A. & Pichancourt, I., (1995).** *'A Human-Machine Interface Reference System for 'EnRoute Air Traffic Control'* EUROCONTROL, EEC Report N$^{o}$. 292, France.

[2] **Derkum, H., Tasset, D., Buck, S., Hooijer, J., & Jackson, A., (1995).** *'Advanced Ground Human Machine Interface for PHARE Demonstration 2'.* EUROCONTROL, PHARE, Brussels.

[3] **PHARE (1998).** '*PD3 GHMI Specification Final Draft',* EUROCONTROL, PHARE, Brussels.

[4] **EATCHIP III Project Team (1997)** *'EATCHIP III Evaluation & Demonstration, Phase II: Simulation Facilities Specification*', EUROCONTROL Experimental Centre, France.

[5] **Denmark Sweden Interface Project Team (1999).** '*DSI Specification ACC/APP'* EUROCONTROL Experimental Centre, France.

[6] **Jackson, A., (1998).** *'HF Integration within Concept development, design and Pre-Operational Evaluation Phases: A Pragmatic approach',* In proceedings of the 3rd EUROCONTROL Human Factors Workshop, '*Integrating Human Factors into the life cycle of ATM systems'*, Luxembourg

[7] **Umbers, I.C., (1976).** *'A Study of Cognitive Skills in Complex Systems'* Unpublished PhD Thesis University of Aston in Birmingham.

[8] **Bainbridge, L., (1983).** '*Ironies of automation*' *Automatica, p775-779.*

[9] **Fitts, P.M., (ed.) (1951).** *'Human Engineering for an Effective Air-Navigation and Traffic Control System',* NRC Committee on Aviation Psychology, Washington D.C.

[10] **Rouse, W.B., (1988).** *'Adaptive Aiding for Human/Computer Control',* Human Factors, 30(4), p87-122.

[11] **RHEA (1998).** *'RHEA, Role of Human in the Evolution of ATM systems*, Final report for DG7, NLR, Amsterdam.

[12] **Erzberger, H., (1988).** *'Automation Tools for Air Traffic Control'* EUROCONTROL Proceedings of the ATC-2000 Seminar, Luxembourg.

[13] **Endsley, Mica R., (1997)**. '*Level of Automation: integrating Humans and Automated Systems',* Human Factors Ergonomics Society 41st Annual meeting

[14] **Norman, D. A. (1990).** *'The "Problem" of Automation: Inappropriate Feedback and Interaction, not Over-automation'.* Paper prepared for Discussion Meeting, "Human Factors in High-Risk Situations", in Philosophical Transactions of the Royal Society of London, B, 1990.

[15] **EUROCONTROL (1998).** *'Eurocontrol Standard Document for On-Line Data Interchange'* Version 2.2, EUROCONTROL, Brussels.

[16] **EUROCONTROL (1997).** *'Operational Requirements for EATCHIP Phase III ATM Added Functions Volume 2: Safety Nets'* Version 1.3, EUROCONTROL, Brussels.

[17] **Shorrock, S.T., Kennedy, R., Isaac, A., Kirwan, B., Andersen, H., Bove, T. (2000).** 'The Human Error in ATM (HERA) Technique' EUROCONTROL, Brussels, in preparation.

[18] **Farley. Todd C., Hansman, R. John, Endsley Mica R., Amonlirdviman, Keith, Vigeant-Langlois, Laurence, (1998).** *'The Effect of Shared Information on Pilot/Controller Situation Awareness and Re-Route Negotiation'* 2nd USA/Europe ATM R&D Seminar, Orlando, Florida, USA

## Autobiographical Notes on the Authors

**Alistair Jackson** graduated in Psychology from the University of Glasgow where he spent 6 years conducting research into visual processing of information on visual displays. From 1980-1993 he provided the human factors input to a multidisciplinary team conducting research into ATM system development for the UK.CAA at the Royal Signals and Research Establishment, Malvern. In 1993 he moved to the EEC where he is currently leader of the CORE Requirements for ATM Working Positions project. He has worked on ODID 2, 3 & 4; was the HMI design lead on PHARE PD1 and participated in PD2 and PD3. He established the EATCHIP III baseline platform at EEC and has since managed a number of projects on behalf of the EATMP Human Resources, Human Factors sub-programme.

**Alexandra Dorbes** received her doctorate in Psychology from the University of Paris–Sorbonne. She has been working in the ATC domain since 1987 contributing to the development of ELECTRA the new French ATC training simulator (by defining roles and designing the graphical user interfaces tools for the different simulation 'actors'). On contracting to the EEC as a Human Factors expert, she participated in the EATCHIP III Operational Concepts Validation programme, producing HMI specifications, and designing and conducting evaluations. She is currently working on the CORE requirements for ATM Working Positions project concerned with the development and improvement of the controller working position.

**Isabelle Pichancour**t received her doctorate in Cognitive Psychology from the University of Paris VIII. She has been involved with ATC as a Human Factors specialist since 1990 when she worked on defining a new graphical HMI for Working Positions for Instructors or Technical Supervisors for the French CAA. Under contract to EUROCONTROL since 1992, she has participated in a variety of study simulation projects; ODID IV and PHARE demonstrations PD1 and PD3, producing HMI specifications, and designing, conducting and analysing experiments.
She currently works on the new project, C3T – Controller Tools and Transition Trials, using a stepped, pragmatic approach to define roles, tasks, working methods and HMI suited to the introduction of ATM decision support tools.