Accepted for publication, Int. J. Human—Computer Studies (1998) 49, 743—766

Formulating the cognitive design problem of Air Traffic Management

John Dowell
Department of Computer Science
University College London

Evolutionary approaches to cognitive design in the air traffic management (ATM) system can be attributed with a history of delayed developments. This issue is well illustrated in the case of the flight progress strip where attempts to design a computer-based system to replace the paper strip have consistently been met with rejection. An alternative approach to cognitive design of air traffic management is needed and this paper proposes an approach centered on the formulation of cognitive design problems. The paper gives an account of how a cognitive design problem was formulated for a simulated ATM task performed by controller subjects in the laboratory. The problem is formulated in terms of two complimentary models. First, a model of the ATM domain describes the cognitive task environment of managing the simulated air traffic. Second, a model of the ATM worksystem describes the abstracted cognitive behaviours of the controllers and their tools in performingw the traffic management task. Taken together, the models provide a statement of worksystem performance, and express the cognitive design problem for the simulated system. The use of the problem formulation in supporting cognitive design, including the design of computer-based flight strips, is discussed.

1. Cognitive design problems

1.1. Crafting the controller's electronic flight strip

Continued exceptional growth in the volume of air traffic has made visible some rather basic structural limitations in the system which manages that traffic. Most clear is that additional increases in volume can only be achieved by sacrificing the 'expedition' of the traffic, if safety is to be ensured. As traffic volumes increase, the complexity of the traffic management problem rises disproportionately, with the result that flight paths are no longer optimised with regard to timeliness, directness, fuel efficiency, and other expedition factors; only safety remains constant. Sperandio (1978) has described how approach controllers at Orly airport switch strategies in order to sacrifice traffic expedition and so preserve acceptable levels of workload. Simply, these controllers switch to treating aircraft as groups (or more precisely, as 'chains') rather than as separate aircraft to be individually optimised.

For the medium term, there is no ambition of removing the controller from their central role in the ATM system (Ratcliffe, 1985). Therefore, substantially increasing the capacity of the system without qualitative losses in traffic management means giving controllers better tools to assist in their decision-making and to relieve their workload (CAA, 1990). Yet curiously, such tools have not appeared in the operational system at large, in spite of sustained efforts made to produce them.

Take the case of the controller's flight progress strip. The strip board containing columns of individual paper strips is the tool which controllers use for planning and as such occupies a more central role in their task than even the radar screen (Whitfield and Jackson, 1982). Development of an electronic strip has been a goal for some two decades (Field, 1985), for the simple reason that until the technical sub-system components have access to the controller's planning, they cannot begin to assist in that planning. Even basic facilities such as conflict detection cannot be provided unless the controller's plans can be accessed and shared (Shepard, Dean, Powley, and Akl, 1991): automatic detection is of limited value to the controller unless it is able to operate up to the extremes of the controller's 'planning horizon' and to take account of the controller's intended future instructions.

Attempts to introduce electronic flight strips, including conflict detection facilities, have often met with rejection by controllers. Rejection has usually been on the grounds that designs either mis-represent the controller's task, or that the benefits they might offer do not offset the increases in cognitive cost entailed in their use. The consistency in this pattern of rejection is of interest since it implicates the approach taken to development.

The approach taken in the United Kingdom has been to develop an electronic system which mimics the structures and behaviours of the paper system. This approach has entailed studies of the technical properties of flight strips, and also their social context of use (Harper, Hughes & Shapiro, 1991), followed by the rapid prototyping of electronic strips designs. But electronic flight strip systems cannot hope to match the physical facility of paper strips for annotation and manipulation, particularly within the work practices of the sector team. Rather, electronic flight strips might only be accepted if their inferior physical properties are compensated by providing innovative functions for actively sharing in the higher level cognitive tasks of traffic management. By actively sharing in tasks such as flight profiling, inter-sector coordinations, etc, electronic flight strips might offset the controller's cognitive costs at higher levels, resulting in an overall reduction in cognitive cost.

These difficulties in the development of the electronic flight strip are symptomatic of the general approach taken to cognitive design within the ATM system. It is an approach which emphasises the value of incremental and evolutionary change. But it is also one which relies, not so much on 'what is known' about the system, as on what is 'tried and tested'. This craft-like approach (Long and Dowell, 1989) has resulted in effective stalemate in respect of the controller's task, since it excludes innovative forms of cognitive design. Without an explicit, complete or coherent analysis of the Air Traffic Management task, the changes resulting from innovative designs cannot be predicted and therefore must be avoided. An alternative approach is needed, and one which offers the required analysis is cognitive engineering, as now discussed.

1.2. Cognitive engineering as formulating and solving cognitive design problems

The development of the ATM system can be seen as an exemplary form of cognitive design problem, one which subsumes a domain of cognitive work (the effective control of air traffic movements) and a worksystem comprising cognitive agents (the controllers) and their cognitive tools (e.g., flight strips). Moreover, it critically includes the effectiveness of that worksystem in performing its work - the actual quality of the air traffic management achieved and the cognitive costs to the worksystem.

Treating air traffic management as a cognitive design problem is consistent with the cognitive engineering approach to development. Cognitive engineering has been variously defined (Hollnagel and Woods, 1983; Norman, 1986; Rasmussen, 1986; Woods and Roth, 1988) as a discipline which can supersede the craft like disciplines of Human Factors and Cognitive Ergonomics. A review of definitions can be found in Dowell and Long (1998). As a discipline, cognitive engineering can be distinguished most generally as the application of engineering knowledge of users, their work and their organisations to solving cognitive design problems. Its characteristic process is one of 'formulate then solve' problems of cognitive design, in contrast with *ad hoc* approaches to improving cognitive systems. Norman (1986) identifies approximation and the systematic trade-off between design decisions as basic features of this process. Ultimately, cognitive engineering seeks engineering principles which can prescribe solutions to cognitive design problems (Norman, 1986; Long and Dowell, 1989).

This paper presents the formulation of the cognitive design problem for a simulated ATM system. To formulate any cognitive design problem takes two starting points (Figure 1). First, there must be some "situation of concern" (Checkland, 1981), in which an instance or class of worksystem is identified as requiring change. In this paper, a simulated ATM system is taken as presenting such a situation of concern (Section 1.4). Second, there must be a conception of cognitive design problems. A conception provides the general concepts, and a language, with which to express particular design problems. Similarly, Checkland (1981) describes how an explicit system model supports the abstraction and expression of problem situations within the soft systems methodology. In this paper, a conception of cognitive design problems proposed by Dowell and Long (1998) supplies the framework for the problem formulation (see Figure 1). That conception is now summarised.

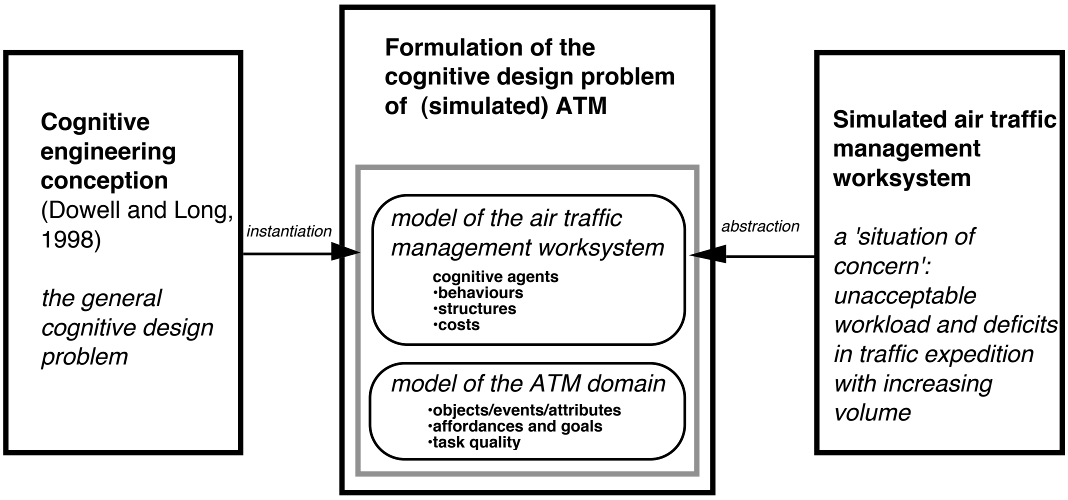

Cognitive engineering conception
(Dowell and Long, 1998)
the general cognitive design problem
instantiation
Formulation of the cognitive design problem of (simulated) ATM
model of the air traffic management worksystem
cognitive agents
•behaviours
•structures
•costs
model of the ATM domain
•objects/events/attributes
•affordances and goals
•task quality
abstraction
Simulated air traffic management worksystem
a 'situation of concern': unacceptable workload and deficits in traffic expedition with increasing volume

Figure 1. Formulation of a cognitive design problem. The problem is abstracted over a simulated ATM system which presents a situation of concern. The problem

formulation instantiates a conception for cognitive engineering .

1.3. Conception of cognitive design problems

Cognitive design problems can be expressed in terms of a dualism of domain and worksystem, where the worksystem is designed to perform work in the domain to some desired level of performance (Dowell and Long, 1998).
Domains might be generally conceived in terms of their goals, constraints and possibilities. Domains consist of *objects* identified by their *attributes*. Attributes emerge at different levels within a hierarchy of complexity within which they are related. Attributes have *states* (or values) and so exhibit an *affordance* for change. Desirable states of attributes we recognise as *goals*. *Work* occurs when the attribute states of objects are changed by the behaviours of a worksystem whose intention it is to achieve goals. However work does not always result in all goals being achieved all of the time, and the variances between goals and the actual outcomes of work are expressed by*task quality* .

The worksystem consists of the cognitive agents and their cognitive tools (technical sub-systems) which together perform work within the same domain. Being constituted within the worksystem, the cognitive agents and their tools are both characterised in terms of structures and behaviours. *Structures* provide the component capabilities for behaviour; most centrally, they can be distinguished as representations and processes. *Behaviours* are the actualisation of structures: they occur in the processing and transformation of representations, and in the expression of cognition in action. There are, therefore, both physical and mental (or virtual) forms of both structures and behaviours. Hutchins (1994) notes that this distinction between structure and behaviour corresponds with a separation of task and algorithm (Marr, 1982); here, however, a task is treated as the conjunction of transformations in a domain and the intentional behaviours which produce those them.

Work performed by the worksystem incurs *resource cost*s. Structural costs are the costs of providing cognitive structures; behavioural costs are the costs of using those structures. Both structural and behavioural costs may be separately attributed to the agents of a worksystem. The *performance* o

f the worksystem is the relationship of the total costs to the worksystem of its behaviours and structures, and the task quality resulting from the decisions made. Critically then, the behaviours of the worksystem are distinguished from its performance (Rouse, 1980) and this distinction allows us to recognise an *economics of performance*. Within this economy, structural and behavioural costs may be traded-off both within and between the agents of the worksystem, and they may also be traded-off with task quality. Sperandio's observations of the Orly controllers, discussed earlier, is an example of the trade-off of task quality for the controller's behavioural costs.
It follows from this conception that the particular cognitive design problem of ATM should be formulated in terms of two models,

- a model of the ATM domain, describing the air traffic processes being managed, and
- a model of the ATM worksystem, describing the agents and technical sub-systems (tools) which perform that management.

These two models are indicated schematically in Figure 1, as the major components of the ATM problem formulation.

1.4. Simulated air traffic management task

The ATM cognitive design problem formulated here is of a simulated ATM system which presents a situation of concern: specifically, the unacceptable increases in workload, and the losses in traffic expedition, with increasing traffic volumes. The simulation reconstructs a form of the air traffic management task. This task is performed by trained subject 'controllers' who monitor the traffic situation and make instructions to the simulated aircraft. The simulation is built on a computational traffic model and provides the common form of ATM control suite (Dowell, Salter and Zekrullahi, 1994). It provides a radar display of the current state of traffic on a sector consisting of the intersection of two en-route airways. It also provides commands via pull-down menus for requesting information from and instructing aircraft (i.e., for interrogating and modifying the traffic model). Last, the control suite includes an inclined rack of paper flight progress strips, arranged in columns by different beacons or reporting points. For each beacon an aircraft will pass on its route through the sector, a strip is provided in the appropriate rack column. The strips tell the controller which aircraft will be arriving when, where, and how (i.e., their height and speed), their route, and their desired cruising height and cruising speed.

Using the radar display and flight strips, the subject controller is able continuously to plan the flights of all aircraft and to execute the plan by making appropriate interventions (issuing speed and height instructions). The subject controller works in a ‘planning space’ in which, reproducing the real system, aircraft must be separated by a prescribed distance, yet should be given flight paths which minimise fuel use, flying time and number of manoeuvres, whilst also achieving the correct sector exit height (Hopkin, 1971). Fuel use characteristics built into the computational traffic model constrain the controller's planning space with regard to expedition, since fuel economy improves with height and worsens with increasing speed. Because of this characteristic, controllers may not solve the planning problem satisfactorily simply by distributing all aircraft at different levels and speeds across the sector. Additional airspace rules (for example, legal height assignments) both constrain and structure the controller's planning space. The controller works alone on the simulation, performing a simplified version of the tasks which would be performed by a team of at least two controllers in the real system; the paper flight strips include printed information which a chief controller would usually add whilst coordinating adjacent sectors.

Increasing volumes of air traffic within this system inevitably result in sacrifices in traffic expedition, if safety is to be maintained. Simply, the traffic management problem (akin to a "game of high speed, 3D chess", Field, 1985) becomes excessively complex to solve. Workload increases disproportionately with additional traffic volumes. The simulated system therefore presents a realistic situation of concern over which a cognitive design problem can be formulated, as now described.

2. Model of the ATM domain

The model of the ATM domain is given in this section. Because of its application to the laboratory simulation, the model makes certain simplifications. For example, the simulation does not represent the wake turbulence of real aircraft, a factor which may significantly determine the closeness with which certain aircraft may follow others; accordingly, the framework makes no mention of wake turbulence. However, the aim here is to present a basic, but essentially correct characterisation of the domain represented by the simulation. Later refinement, by the inclusion of wake turbulence for instance, is assumed to be possible having established the basic characterisation.

2.1 Airspace objects, aircraft objects, and their dispositional attributes

An instance of an ATM domain arises in two classes of elemental objects: airspace objects, and aircraft objects, defined by their respective attributes. Aircraft objects are defined by their callsign attribute and their type attributes, for example, laden weight and climb rate. Airspace objects include sector objects, airway interval objects, flight level objects, and beacon objects. Each is defined by their respective attributes, for example, beacons by their location. Importantly, the attributes of aircraft and airspace objects have an invariant but necessary state with respect to the work of the controller: these kinds of attribute we might call 'dispositional' attributes.

2.2 Airtraffic events and their affordant attributes

Notions of traffic intuitively associate transportation objects with a space containing them. In the same way, an instance of an ATM domain defines a class of airtraffic events in the association of airspace objects with aircraft objects at particular instants. Airtraffic events are, in effect, a superset of objects, where each object exists for a defined time. They have attributes emerging in the association of aircraft objects with airspace objects; these minimally include the attributes of:
Position (given by airway interval object currently occupied)
Altitude (given by flight level (FL) object currently occupied)
Speed (given in knots, derived from rate of change in Position and Altitude)
Heading (given by next routed beacon object(s))
Time (standard clock time)
Unlike the dispositional attributes of airspace and aircraft objects, PASHT attributes of airtraffic events have a variable state determined by the interventions of the controller; they might be said to be ‘affordant’ attributes.

2.3 Airtraffic event vectors and their task attributes

Each attribute of an airtraffic event can possess any of a range of states; generally, each attribute affords transformation from one state to another. However there is an obvious temporal continuity in the ATM domain since time-ordered series of airtraffic events are associated with the same aircraft. Such a series we can describe as an 'airtraffic event vector'. Whilst event vectors subsume the affordant attributes (the PASHT attributes) of individual airtraffic events, they also exhibit higher level attributes. The task of the controller arises in the transformation of these 'task attributes' of event vectors.

The two superordinate task attributes of event vectors are safety and expedition. Safety is expressed in terms of a 'track separation' and a vertical separation. Track separation is the horizontal separation of aircraft, whether in passing, crossing or closing traffic patterns, and is expressed in terms of flying time separation (e.g., 600 seconds). A minimum legal separation is defined as 300 seconds, and all separations less than this limit are judged unsafe. Aircraft on intersecting paths but separated by more than the legal minimum are judged to be less than safe, and the level of their safety is indexed by their flying time separation. Aircraft not on intersecting paths (and outside the legal separation) are judged to be safe. A legal minimum for vertical separation of one flight level (500m) is adopted: aircraft separated vertically by more than this minimum are judged to be safe, whilst a lesser separation is judged unsafe.

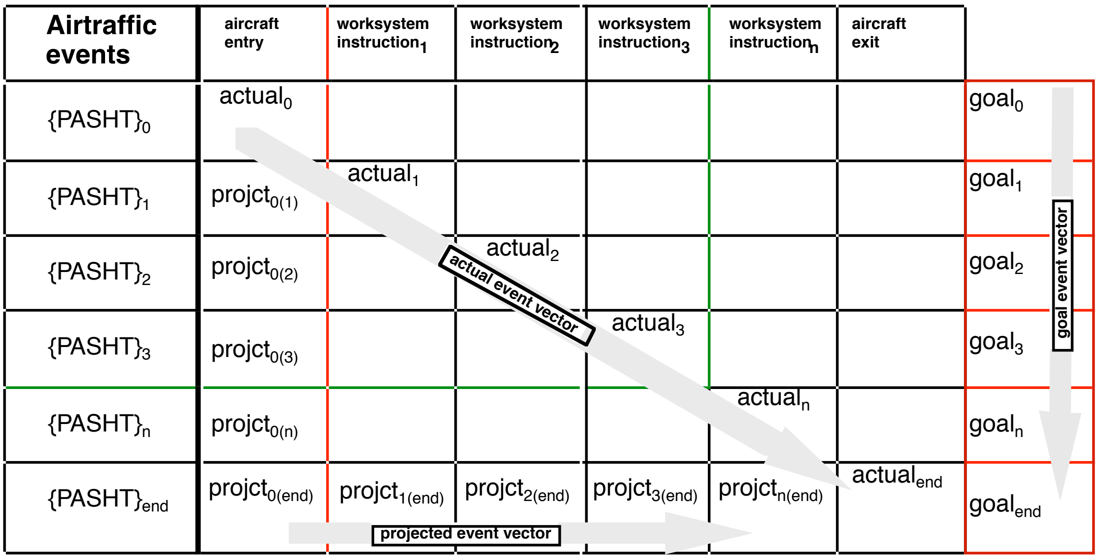
Expedition subsumes the task attributes of:

- flight progress', that is, the duration of the flight (e.g., 600 seconds) from entry onto the sector to the present event ;
- 'fuel use', that is, the total of fuel used (e.g., 8000 gallons) from entry onto the sector ;
- number of manoeuvres that is, the total number of instructions for changes in speed or navigation issued to the aircraft from entry onto the sector; and
- 'exit height variation', that is, the variation (eg, 1.5 FLs) between actual and desired height at exit from the sector.

Three different sorts of airtraffic event vector can be defined: actual; projected and goal. Each vector posseses the same classes of task attribute, but each arises from different air traffic events. Figure 2 schematises the three event vectors within an event vector matrix.

- First, the **actual event vector** describes the time-ordered series of actual states of airtraffic events: in other words, how and where an aircraft *was* in a given period of its flight. Aircraft within the same traffic scenario can be described by separate, but concurrent actual event vectors. Figure 2 schematises an actual event vector (actual_0 ... actual_n, actual_{end}) related to the underlying sequence of air traffic events (PASHT values). For example, actual_1 represents the actual task attribute values for a given aircraft at the first instruction issued by the controller to the airtraffic. It expresses the actual current safety of a particular aircraft, the current total of fuel used, the current total of time taken in the flight, and the current total of manoeuvres made. Exit height variation applies only to the final event (actual_{end}) in the event vector, when the final exit height is determined.
- Second, the **goal event vector** describes the time-ordered series of *goal* states of airtraffic events: in other words, how and where an aircraft *should have been* in a given period of its flight. Figure 2 schematises a goal event vector (goal_0 ... goal_n, goal_{end}) within the event vector matrix. For example, goal_1 represents the goal values of the task attributes at the controller’s first intervention, in terms of the goal level of safety (i.e., the aircraft should be safe), and current goal levels of fuel used, time taken, and number of manoeuvres made. These values can be established by idealising the trajectory of a single flight made across the sector in the absence of any other aircraft, where the trajectory is optimised for fuel use and progress. The goal value for exit height variation applies only to the final event (goal_{end}).

- Third, the **projected event vector** describes the time-ordered series of projected future states of airtraffic events: in other words, how and where an aircraft *would have been* in a given period of its flight, given its current state - and assuming no subsequent intervention by the controller (an analysis commonly provided by 'fast-time' traffic simulation studies). In practice, only the projected exit state, and projected separation conflicts at future intermediate events, are needed for the analysis, and only from the start of the given period and at each subsequent controller intervention. In this way, the potentially large number of projected states is limited. Figure 2 schematises a projected event vector ($\mathrm{projct}_{0(end)}$, .. $\mathrm{projct}_{n(end)}$) within the event vector matrix. For example, $\mathrm{projct}_{1\,(end)}$ represents the projected end values of the task attributes following the controller's first intervention. It describes the projected final safety state of the aircraft, total projected fuel use for its flight through the sector, its total projected flight time through the sector, the total number of interventions and the projected exit height variation.



Airtraffic events	aircraft entry	worksystem instruction$_1$	worksystem instruction$_2$	worksystem instruction$_3$	worksystem instruction$_n$	aircraft exit	
$\{PASHT\}_0$	actual_0						goal_0
$\{PASHT\}_1$	$\mathrm{projct}_{0(1)}$	actual_1					goal_1
$\{PASHT\}_2$	$\mathrm{projct}_{0(2)}$		actual_2				goal_2
$\{PASHT\}_3$	$\mathrm{projct}_{0(3)}$			actual_3			goal_3
$\{PASHT\}_n$	$\mathrm{projct}_{0(n)}$				actual_n		goal_n
$\{PASHT\}_{end}$	$\mathrm{projct}_{0(end)}$	$\mathrm{projct}_{1(end)}$	$\mathrm{projct}_{2(end)}$	$\mathrm{projct}_{3(end)}$	$\mathrm{projct}_{n(end)}$	actual_{end}	goal_{end}

Figure 2. The event vector matrix

An event vector matrix of this form was constructed for each of the controller subjects performing the simulated air traffic management task. It was constructed in a spreadsheet using a protocol of aircraft states and controller instructions collected by the computational traffic model.

The differentiation of actual, goal and projected event vectors now enables expression of the quality of air traffic management by the controller.

2.4 Quality of air traffic management (ATMQ)

The final concept in this framework for describing the ATM domain is of task quality. Task quality describes the actual transformation of domain objects with respect to goals (Dowell and Long, 1998). In the same way, the Quality of Air Traffic

Management (ATMQ) describes the success of the controller in managing the air traffic with regard to its goal states.

ATMQ subsumes the Quality of Safety Management (QSM) and Quality of Expedition Management (QEM). Although there are examples (Kanafani, 1986) of such variables being combined, here the separation of these two management qualities is maintained. Since expedition subsumes the attributes of fuel use, progress, exit conditions and manoeuvres, each of these task attributes also has a management quality. So, QEM comprises:

- QFM: Quality of fuel use management
- QPM: Quality of progress management
- QXM: Quality of exit conditions management
- QMM: Quality of manoeuvres management

These separate management qualities are combined within QEM by applying weightings according to their perceived relative salience (Keeney, 1993).

A way of assessing any of these traffic management qualities would be (following Debenard, Vanderhaegen and Millot, 1992) to compare the actual state of the traffic with the goal state. But such an assessment could not be a true reflection of the controller’s management of the traffic because air traffic processes are intrinsically goal directed and partially self-determining. In other words, each aircraft can navigate its way through the airspace without the instructions of the controller, each seeking to optimise its own state; yet because each is blind to the state and intentions of other aircraft, the safety and expedition of the *airtraffic* will be poorly managed at best. ATMQ then, must be a statement about the 'added value' of the controller's contributions to the state of a process inherently moving away or towards a desired state of safety and expedition. To capture this more complex view of management quality, ATMQ must relate the actual state of the traffic relative to the state it would have had if no (further) controller interventions had been made (its projected state) and relative to its goal state. In this way, ATMQ can be a measure of gain attributable to the controller.

Indices for each of the management qualities included in ATMQ can be computed from the differences between the goal and actual event vectors. The form of the index is such that the quality of management is optimal when a zero value is returned, that is to say, when actual state and goal state are coincident. A negative value is returned when traffic management quality is less than desired (goal state). For, QPM and QFM, a value greater than zero is possible when actual states are better than goal states, since it is possible for actual values of fuel consumed or flight time to be less than their goal values. Further, by relating the index to the difference between the goal and projected event vectors, the significant difference of the ATM worksystem's interventions over the scenario are given. In this way, the 'added value' of the worksystem's interventions is indicated.

Two forms of ATMQ are possible by applying the indices to the event vector matrix (Figure 2). Both forms will be illustrated here with the data obtained from the controller subjects performing the simulated ATM task. The analysis of ATMQ is output from the individual event vector matrices constructed in spreadsheets, as earlier explained.

The first form of ATMQ describes the task quality of traffic management over a complete period. It describes the sum of management qualities for all aircraft over their flight through the sector and so can be more accurately designated ATMQ(fl) to identify it as referring to completed flights. It is computed by using the initially projected, goal and actual final attribute values ($\text{projct}_{0(\text{end})}$, goal_{end}, $\text{actual}_{\text{end}}$) for each event vector (i.e., the 'beginning and end points'). The functions by which these ATMQ(fl) management qualities are calculated are given in Appendix 1.

Figure 3 illustrates the assessment of ATMQ(fl) - in other words the assessment of management qualities over completed flights for the controllers separately managing the same traffic scenario. The scenario consisted of six aircraft entering the sector over a period of 45 minutes. ATMQ(fl) was first computed for each form of management quality, for each aircraft under the control of each controller. Figure 3 presents a summation of this assessment for each of the controllers for each of the five management qualities but for all six individual flights. For example, we are able to see the quality with which Controller 1 managed the safety (i.e., QSM) and fuel use (i.e., QFM) of all six aircraft under her control over the entire period of the task.

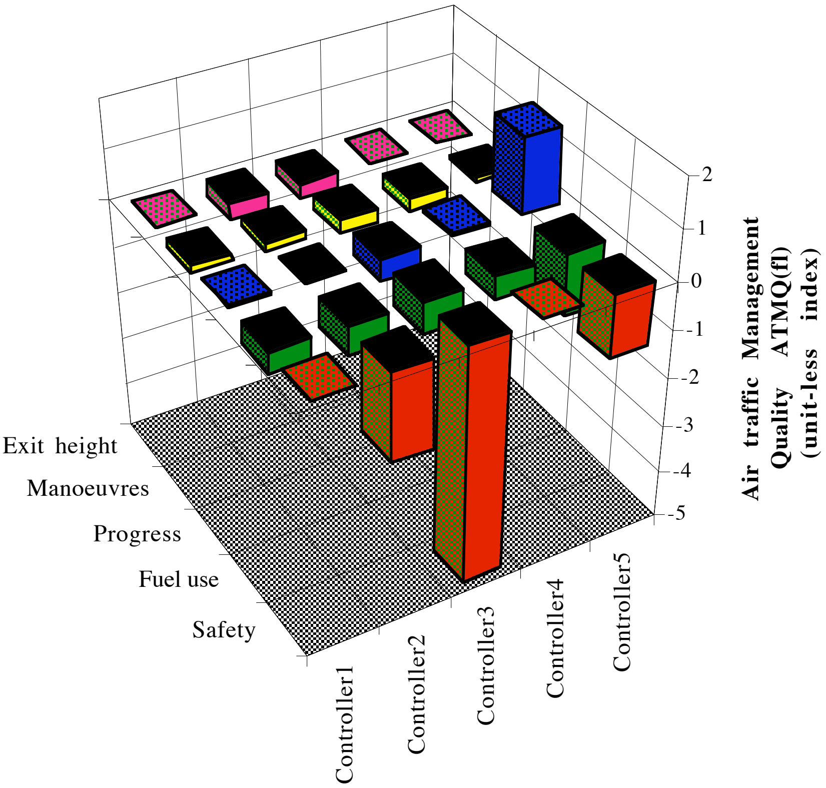


Figure 3. Assessment of Air Traffic Management Quality for all completed flights of each controller.

It is important to note that ATMQ(fl) is achronological, in so much that it describes the quality of management of each flight after its completion: hence, it would return the same value whether all aircraft had been on the sector at the same time during the scenario, or whether only one flight had been on the sector at any one time. Whilst this kind of assessment provides an essential view of the acquittal of management work from the point of view of each aircraft, it provides a less satisfactory view of the acquittal of management work from the point of view of the worksystem.

The second form of ATMQ describes the task quality of traffic management for each intervention made by an individual controller. This second kind of task quality is designated as ATMQ(int), to identify it as referring to interventions and is computed from the currently projected end state, previously projected end state, and new goal end state (for example, $\mathrm{projct}_{1(end)}$, $\mathrm{projct}_{2(end)}$, goal_{end} for the second intervention). The functions by which these ATMQ(int) management qualities are calculated are given in Appendix 2.

Figure 4 illustrates this second principal form of ATMQ - the assessment of ATMQ(int) for all aircraft with each intervention of an individual controller. For the sake of clarity, only the qualities of safety (QSM), fuel use (QFM) and progress (QPM) are shown. For each management intervention made by the controller during the period of the task, these three management qualities are described, each triad of data points relating to an instruction issued by the controller to one of the six aircraft.

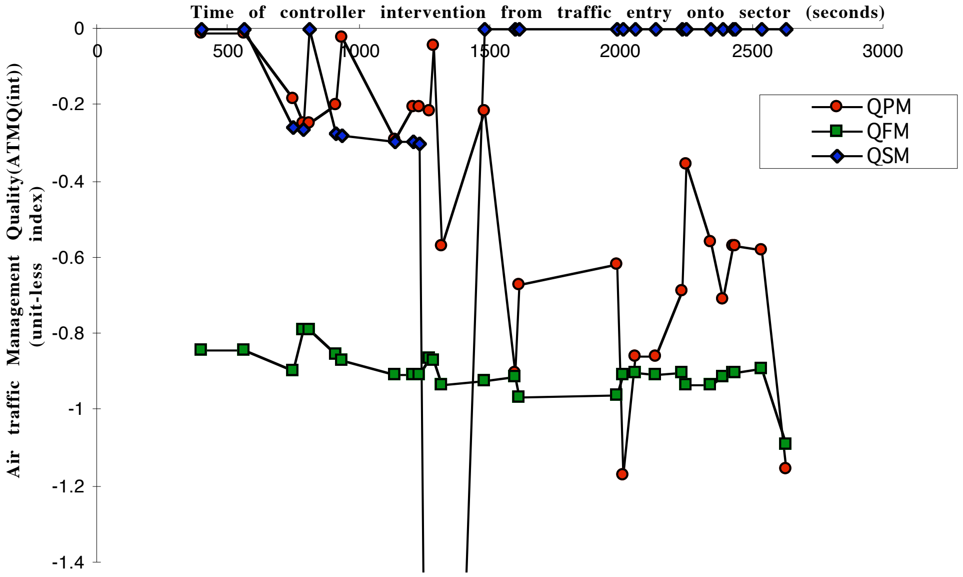


Figure 4. Qualities of: progress management (QPM); fuel use management (QFM); and safety management (QSM) achieved by Controller3 during the task.

Finally, although the analysis of ATMQ requires the worksystem's interventions to be explicit, it does not require that there actually be any interventions. After all, when no problems are present in a process, good management is that which monitors but makes no intervention. Similarly, if the projected states of airtraffic events are the same as the goal states, then good management is that which makes no interventions, and in this event, ATMQ would return a value of zero.

To summarise, the ATM domain model describes the work performed in the Air Traffic Management task. It describes the objects, attributes, relations and states in this class of domain, as related to goals and the achievement of those goals. The model applies the generic concepts of domains given by the cognitive engineering conception presented earlier. The model describes the particular domain of the simulated ATM task from which derives the example assessments of traffic management quality given here. Corresponding with the domain model, the worksystem model presented in the next section describes the system of agents that perform the Air Traffic Management task.

3. Model of the ATM worksystem

A model of the worksystem which performs the Air Traffic Management task can be generated directly from the domain model. The representations and processes minimally required by the worksystem can be derived from the constructs which make up the domain model. In this way, ecological relations (Vera and Simon, 1993) bind the worksystem model to the domain. Woods and Roth (1988) identify the ecological modelling of systems as a central feature of cognitive engineering, given the concern for designing systems in which the cognitive resources and capabilities of users are matched to the demands of tasks.

The ecological approach to modelling worksystems has been contrasted (Payne, 1991) both with the architecture-driven and the phenomenon-driven approaches: that is to say, it can be contrasted both with the deductive application of general architectures to models of specific behaviours (Howes and Young, 1997), and with attempts to generate 'local' models from empirical observations of specific performance issues. However this distinction is too sharply drawn and needs to be further qualified, since the organisation of a worksystem model (as opposed to the content), is not determined by the domain model. First, the ATM worksystem model instantiates the conception of cognitive design problems. Hence the concepts of structure, behaviour and costs, are used as a primary partitioning of the ATM worksystem model. Second, the ATM worksystem model adopts specific constructs from the blackboard architecture (Hayes-Roth and Hayes-Roth, 1979) to organise the particular relations between the representations and processes deriving from the ATM domain model. Hence a general cognitive architecture is employed selectively in the ATM worksystem model.

3.1 Structures of the ATM worksystem

The structures of the ATM worksystem consist, at base, of representations and processes. The representations constructed and maintained by the ATM worksystem are shown schematically in Figure 5, contained within a blackboard of airtraffic events, a blackboard of event vectors, and a schedule of planned instructions.

The blackboard of airtraffic events contains a representation of the *current airtraffic event* (e_1)constructed from sensed traffic data. The blackboard has two dimensions, a real time dimension and a dimension of hypotheses about the PASHT attribute states of individual aircraft. Knowledge sources associated with this blackboard support the construction of hypotheses about the attributes of airtraffic events. For example, knowledge sources concerning the topology of the sector airways support the construction of hypotheses about heading attributes. As the ATM worksystem monitors flights through the sector, it maintains a representation of a succession of discrete airtraffic events.

A blackboard of event vectors contains separate representations of a current event vector, a goal event vector, and a planned vector. The *current event vector* expresses the actual values of task attributes deriving from the current airtraffic event, and the projected values of those task attributes at future events. A representation of the *goal event vector* expresses the goal values of task attributes for the current and projected airtraffic events. A representation of a *planned event vector* expresses planned values of task attributes for the current and projected airtraffic events. Critically, this vector is distinct from the goal event vector, allowing that the planned state of the traffic will not necessarily coincide with the idealised goal state.

The blackboard of event vectors has two dimensions, a real time dimension and a dimension of hypotheses about the task attributes of event vectors. The hypotheses then concern the attributes of safety and expedition of each vector, where the attribute of expedition subsumes the individual attributes of progress, fuel use, number of manoeuvres and exit height variation. Knowledge sources separately associated with this blackboard support the construction of hypotheses about the attributes of event vectors. For example, knowledge sources about the minimum legal separations of traffic, and about aircraft fuel consumption characteristics, support the construction of hypotheses about safety and fuel use, respectively. Other knowledge sources support the ATM worksystem in reasoning about differences between the current vector and goal vector, and in constructing the planned vector.

Apparent within the blackboard of event vectors are a distinct monitoring horizon and planning horizon. The current event vector extends variably into future events. The temporal limits of the current vector constitute a 'monitoring horizon' of the ATM worksystem: it is the extent to which the worksystem is 'looking ahead' for traffic management problems. Similarly the planned event vector extends variably into the future events. Its temporal limits constitute a 'planning horizon': it is the extent to which the ATM worksystem is 'planning ahead' to solve future traffic management problems. Both monitoring horizon and planning horizon can be expected to be reduced with increasing traffic volumes and complexities.

The planned vector is executed by a set of planned instructions. Planned instructions are generated by reasoning about the set of planned vectors for individual aircraft and the options for possible instructed changes in speed, heading or altitude. This reasoning is again supported by specialised knowledge sources. The worksystem maintains a schedule of planned instructions, shown in Figure 5 as a separate representation: instruction i_1 is shown executed at time t_1.

The complexity of the representations of the ATM worksystem is complemented by the simplicity of its processes. Two kinds of abstract process are specified: generation processes and, evaluation processes and can address the event-level and the vector-level representations. Two kinds of physical process are specified addressed to the event-level representations: monitoring processes and executing processes.

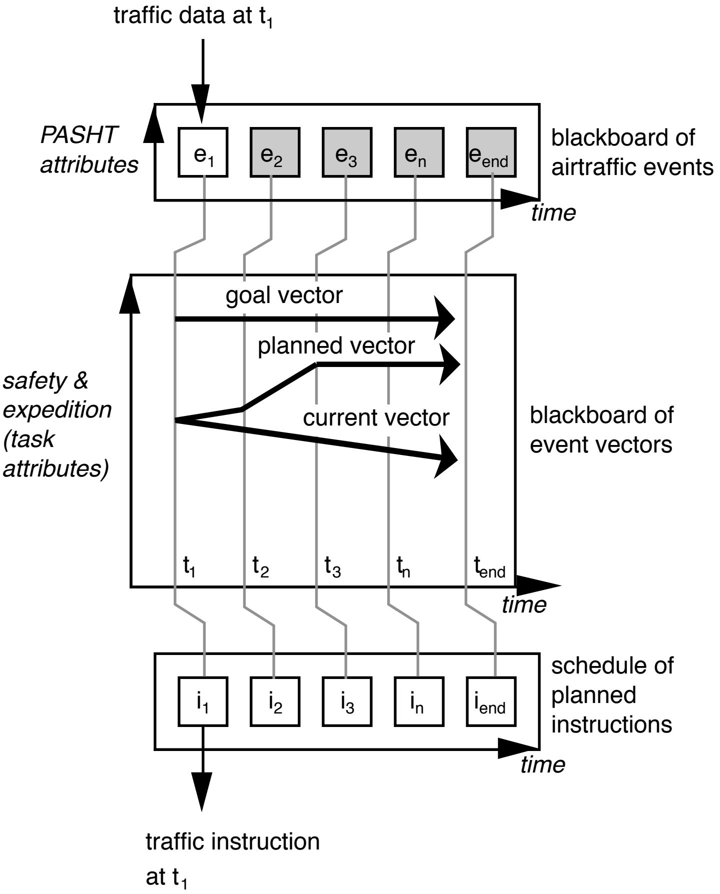


Figure 5. Schematic view of representations maintained by the ATM worksystem

3.2 Behaviours of the ATM worksystem

The behaviours of the ATM worksystem are the activation of its structures, both physical and abstract, which occurs when the worksystem is situated in an instance of an ATM domain. Behaviours, whether physical or abstract, are understood as the processing of representations, and so can be defined in the association of processes with representations. Eight kinds of ATM worksystem behaviour can be defined, grouped in three super-ordinate classes of monitoring, planning and controlling (i.e., executing) behaviours:

Monitoring behaviours

- Generating a current airtraffic event.
  The ATM worksystem generates a representation of the current airtraffic event. This behaviour is a conjunction of both monitoring and generating processes addressing the monitoring space. The representation which is generated expresses values of the PASHT attributes of the current airtraffic event.

- Generating a current event vector .
  The ATM worksystem generates a representation of the current vector by abstraction from the representation of the current airtraffic event. The

representation expresses current actual values, and currently projected values of the task attributes of the event profile. In other words, it expresses the actual and projected safety and expedition of the traffic.

- Generating a goal event vector.
  The representation of the goal vector is generated directly by a conjunction of monitors and generators. The representation expresses goal values of the task attributes of the event profile.

- Evaluating a current event vector.
  The ATM worksystem evaluates the current vector by identifying its variance with the goal vector. This behaviour attaches 'problem flags' to the representation of the current vector.

Planning behaviours

- Generating a planned event vector.
  If the evaluation of the current vector with the goal vector reports an acceptable conformance of the former, then the current vector is adopted as the planned vector. Otherwise, a planned vector is generated to improve that conformance.

- Evaluating a planned event vector.
  With the succession of current vector representations, and their evaluation, the ATM worksystem evaluates the planned vector and a new planned vector is generated.

- Generating a planned instruction.
  Given the planned vector, the instructions needed to realise the plan will be generated by the ATM worksystem, and perhaps too, the actions needed to execute those interventions.

Controlling behaviour

- Executing a planned intervention.
  The ATM worksystem generates the execution of planned interventions, in other words, it decides to act to issue an instruction to the aircraft.

These eight worksystem behaviours can be expressed continuously and concurrently. With the changing state of the domain, not least as a consequence of the worksystem's interventions, each representation will be revised.

3.3 Cognitive costs

Cognitive costs can be attributed to the behaviours of the ATM worksystem and denote the cost of performing the air traffic management task. These cognitive costs are a critical component of the performance of the ATM worksystem, and so too of this formulation of the ATM cognitive design problem. Cognitive costs are derived from a model of the eight classes of worksystem behaviour as they are expressed over the period of the air traffic management. The model of worksystem behaviours is established using a post-task elicitation method, as now described.

Following completion of the simulated traffic management task, the controller subject was required to re-construct their behaviour in the task by observing a video recording of traffic movements on the sector during the task. The recording also showed all requests the controller had made to aircraft for height and speed information, and it showed the instructions that were issued to each aircraft. A set of unmarked flight strips for the traffic scenario was provided. As the video record of the task was replayed, the controller was required to manipulate the flight strips in the way they would have done during the task. For example, as each aircraft entered the sector they were required to move the appropriate strip to the live position. As the aircraft progressed through the sector, its sequence of strips would be 'made live' and then discarded. The controller annotated the flight strips with information obtained from each aircraft request made during the task, and with each instruction issued. The controller was required to view the videotape as a sequence of five minute periods. They were able to halt the tape at any point, for example, in order to update the flight strips. However, no part of the videotape could be replayed.

At the end of each five minute period, the controller was required to complete a 'plan elicitation' sheet. The plan elicitation sheet required the controller to state for each aircraft, the interventions they were planning to make. The specific planned instruction was to be stated (height or speed change) as well as the location of the aircraft when the instruction would be issued. The controller was asked to identify aircraft for which, at that time, no interventions were planned, whether because consideration had not then been given to that aircraft, or a decision had been made that no further instructions would be needed. When the sheet was completed it was set to one side and the controller then viewed the next five minute period of the videotape, after which they completed a new plan elicitation sheet. In this way, for each aircraft at the end of each five minute interval, all planned interventions were described.

This elicited protocol of sampled planned interventions was then compared with the instructions originally issued, as recorded by the traffic model. The comparison indicated a number of issued instructions whose plan had not been reported in the corresponding previous sampling interval of the post-task elicitation. These additional instructions were taken to indicate planning behaviours wherein a planned intervention had been generated and executed between elicitation points. Hence, the record of executed interventions was used to augment and further complete the record of planned interventions obtained from the post-task elicitation. The result of this analysis was a data set describing the sequence of planned interventions for each aircraft over the period of the traffic management task.

The analysis was continued by abstracting the classes of planned interventions for each aircraft over the scenario, divided again into a succession of five minute intervals. Four different kinds of planned intervention were identified:

(i) interventions planned at the beginning of an interval and not executed within the interval.

(ii) planned interventions which were a revision of earlier plans, but which also were not executed within the five minute interval.

(iii) planned interventions which were also executed within the same five minute interval, plans executed exactly, and plans revised when executed.

(iv) plans for interventions made during the five minute interval, but where those plans were not described at all at the beginning of the interval.

Each of these intervention plans was identified by its instruction type, that is, whether it was a planned instruction for a height or speed change.

Representations of airtraffic events, planned event vectors, current vectors and goal vectors are implicit in the analysis of planned interventions. These representations were inferred from the analysis of planned interventions by applying a set of eight rules deriving from the ATM worksystem model, as given in Table 1.

1) the behaviour of generating a representation of the current airtraffic event was associated with any planned intervention for a given aircraft within a given interval, whether reported or inferred, except where those planned interventions were (a) reported rather than inferred, and (b) a reiteration of a previous reporting of a planned intervention, and (c) not executed within the interval.

2) the behaviour of generating a representation of the current vector was only associated with those planned interventions already associated with the behaviour of generating an event representation, except where (a) the planned intervention is a revision of an earlier planned intervention (b) and the planned intervention is not executed within the same interval.

3) the behaviour of generating a goal event vector was only associated with the first planned intervention for each aircraft.

4) the behaviour of evaluating the current vector was associated with all planned interventions already associated with a behaviour of generate a current vector.

5) the behaviour of generating a planned vector was associated with all planned interventions already associated with a behaviour of evaluating a current vector.

6) the behaviour of evaluating a planned vector was associated only with planned interventions which were revisions of earlier planned interventions, regardless of whether they were reported or inferred.

7) the behaviour of generating a planned intervention was associated with all planned interventions already associated with a behaviour of generating a planned vector, or where the planned intervention was a revision of an earlier reported planned intervention

8) the behaviour of generating the execution of a planned intervention was identified directly from the model of planned interventions

Table 1. Rules applied to constructing the worksystem behaviour model from the analysis of planned interventions.

The result of this analysis is a model of the eight cognitive behaviours of the ATM worksystem expressed over the period of the task. Cognitive costs can be derived from this model by applying the following simplifying assumptions. First, costs are atomised, wherein a cost is separately associated with each instance of expressed behaviour. Second, a common cost 'unit' is attributed to each such instance. Two different but complementary kinds of assessment of behavioural cognitive costs are possible. A cumulative assessment describes the cognitive costs associated with each class of behaviour over the complete task, based on the total number of expressed instances of this class of behaviour. A continuous assessment describes the cognitive costs associated with each class of behaviour over each interval. The metric used in both forms of assessment is simply the number of instances of expressed behaviour in a specific class.

An example of the cumulative assessment of the controller's behavioural costs is given in Figure 6. The figure presents the behavioural costs of each class of controller behaviour exhibited during the traffic management task.

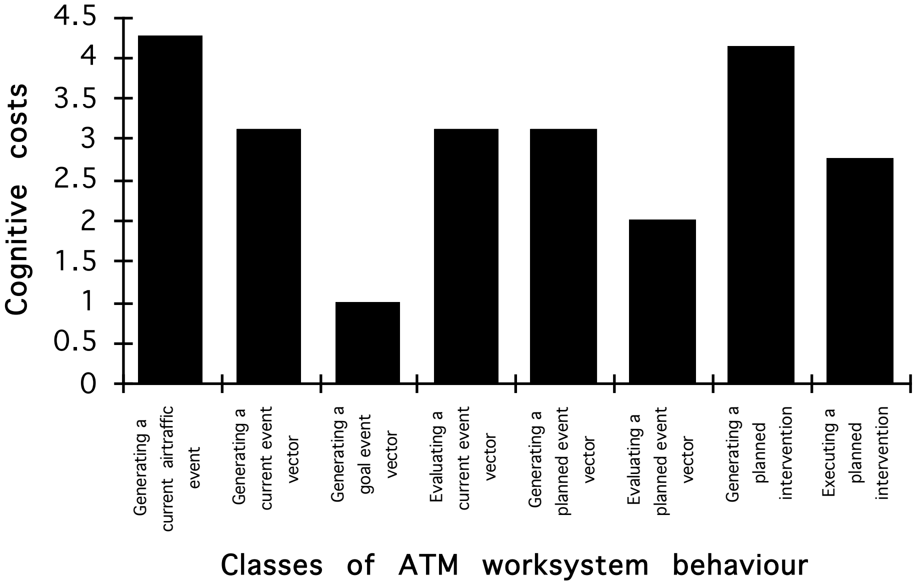


Figure 6. Cumulative assessment of cognitive costs for each class of ATM worksystem behaviour

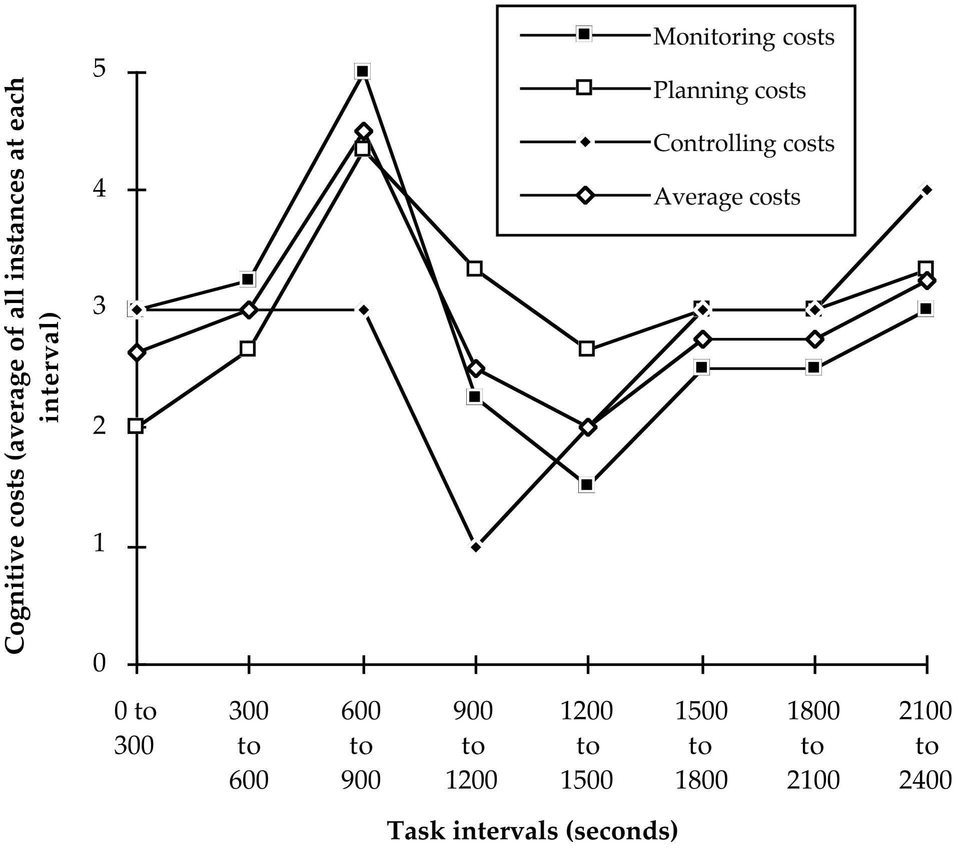


Figure 7. Continuous assessment of cognitive behavioural costs.

Examining the variation across categories, the costs of generating goal vectors were less than any other category. The costs of generating a representation of the current event, and the costs of generating planned interventions, were greater than any other category. Other categories of behaviour incurred seemingly equivalent levels of cost. In terms of the superordinate categories of behaviour, the cognitive costs of planning appear equivalent to those of monitoring and controlling.

An example of the continuous assessment of the same controller's behavioural costs is given in Figure 7. It is an assessment of all classes of cost over each sampling interval (300 seconds) of the task. For simplicity, this assessment is presented as the costs of the superordinate classes of behaviour of monitoring, planning and controlling over each interval. Again, the assessment is produced directly from the number of expressed instances of each class of worksystem behaviour. The continuous assessment includes the average across all costs over each interval.

The continuous assessment suggests that costs rose from the first five minute interval of the task to reach a maximum in the third interval. Because all the aircraft had arrived on the sector by the third interval in the task, the increase in cognitive behavioural costs might be interpreted as the effect of traffic density increases. However, since costs then fall to a minimum in the fifth interval, this interpretation is implausible. Rather, the effect is due to an increase then decrease in monitoring and planning costs as the controller monitored the entry of each aircraft and generated a plan. Although the plan might later be modified, planning behaviours would predominate in the first part of the task. The plan would later be executed by the worksystem's controlling behaviours, and indeed, Figure 7 indicates that the cognitive costs of controlling behaviours predominated over both monitoring and planning costs in the final interval of the task.

The simplifying assumptions adopted in this analysis of cognitive costs need to be independently validated before the technique could be exploited more generally. They can be seen as an example of the approximation which Norman associates with Cognitive Engineering, and which allows tractable formulations of complex problems. As an assessment of cognitive costs based on a model of cognitive behaviour, the analysis contrasts with current methods for assessment of mental workload applied to the ATM task, methods which include concurrent self-assessment by controllers on a four point scale, and other assessments based on observations of the number and state of flight strips in use on the sector suite. Within the primary aim of this paper, the analysis exemplifies the incorporation of cognitive costs within the formulation of the cognitive design problem of ATM.

4. Using the problem formulation in cognitive design

Taken together, the models of the ATM domain and ATM worksystem provide a formulation of the cognitive design problem of Air Traffic Management. The domain model describes the work of air traffic management in terms of objects and relations, attributes and states, goals and task quality (goal achievement). The

worksystem model describes the system that performs the work of air traffic management, in terms of structures, processes and the costs of work.
The models have been illustrated with data captured from a simulated ATM system, wherein controller subjects performed the simulated management task with a computational traffic model.

In the case of the simulated system, the data indicate a worksystem which achieves an insufficient level of traffic management quality and incurs an undesirable level of cognitive cost. The assessment of ATMQ (fl) for all controllers indicated, for example, an inconsistent management of traffic safety (Figure 3). The assessment of ATMQ (int) for Controller 3 indicated, for example, a declining management of progress over the period of the task, and a sub-optimal trade-off between management of progress and of fuel use (Figure 3). Cognitive costs associated with specific categories of behaviour having a level significantly higher than average might also be considered undesirable, such as the category of generating a planned intervention (Figure 6). These data express the requirement for a revised worksystem able to achieve an acceptable trade-off (Norman, 1986) between task quality and cognitive costs.

Because it expresses this cognitive requirement, the problem formulation has the potential to contribute to the specification of requirements for worksystems. Cognitive requirements should be seen as separate from, and complimentary to, software systems requirements. Both kinds of requirement must be met in the design of software-intensive worksystems. Such an approach would mark a shift from standard treatments of software systems development (Sommerville, 1996) wherein users' tasks and capabilities are interpreted as and re-expressed as 'non-functional' requirements of the user interface of the software system.

As well as supporting the specification of requirements, the formulation of the ATM cognitive design problem may also be expected to support the design of worksystems. We might, for example, consider how the problem formulation can contribute to the design of an electronic flight progress strip, earlier described as a focal issue in the development of a more effective ATM system. The problem formulation provides a network within which the flight progress strip can be understood in terms of what it is, and how it is used. First, the domain model allows analysis of the flight strip as a representation. For example, each paper flight strip represents a specific airtraffic event of an aircraft passing a particular beacon. It also represents for reference purposes the preceding and the following such events. The printed information on the strip describes the goal attributes of this airtraffic event in terms of desired height, speed and heading. The controller's annotations of the strip describe both instructions issued and planned instructions. Hence the strip provides a representation of PASHT attributes of the given event. The strip does not represent event vectors, or their task attributes. The worksystem model tells us that the controller must construct the current, goal and planned event vectors, and their attributes, from the PASHT level representation on the strip. These examples indicate how the problem formulation can begin to be used to describe the flight strip and to reason about how the strip is used.

The problem formulation supports the process of evaluation, including the formative evaluation of specific design defects. For example, Controller3 achieved a poor management of safety (QSM) over the period of the task (see Figure 4) due to three interventions made some 1250 seconds into the task. The domain model indicated that the first of the three instructions was for one aircraft to climb above and behind another aircraft, leading to a separation infringement. The worksystem model, constructed from the post-task protocol analysis, described the plans that lead to this misjudgement.

To conclude at a discipline level, the problem formulation presented in this paper can be viewed more generally in terms of the claimed emergence of cognitive engineering. Dowell and Long (1998) have identified design exemplars as a critical entity in the discipline matrix of cognitive engineering. An exemplar is a problem formulation and its solution. Exemplars exemplify the use of cognitive engineering knowledge in solving problems of cognitive design; and they serve as cases for reasoning about new problems. By contrast, craft practices of cognitive design use demonstrators and 'design classics' as its exemplars, a role occupied, for example, by the Macintosh graphical user interface. By contrast, the exemplars of cognitive engineering must be abstractions : they must be formulations of design problems and solutions. The formulation in this paper of the ATM cognitive design problem is an attempt to better understand and advance the construction of exemplars for cognitive engineering.

Acknowledgement. This work was conducted at the Ergonomics and HCI Unit, University College London. I am indebted to Professor John Long for his critical contributions.

References

Checkland P., 1981. *Systems thinking, systems practice*. John Wiley and Sons: Chichester.

Debenard S., Vanderhaegen F., and Millot. P., 1992. An experimental investigation of dynamic allocation of tasks between air traffic controller and AI system. In *Proc. of 5th symposium 'Analysis, design and evaluation of man machine systems,* The Hague, Holland, June 9-11.

Dowell J. and Long. J.B., 1998. Conception of the cognitive engineering design problem, *Ergonomics*. vol 41, 2, pp. 126 - 139.

Dowell J., Salter I. and Zekrullahi S., 1994. A domain analysis of air traffic management work can be used to rationalise interface design issues. In Cockton G., Draper S. and Weir G. (ed.s),*People and Computers IX*, CUP.

Field A., 1985. *International Air Traffic Control*. Pergamon: Oxford.

Harper R. R., Hughes J. A, and Shapiro D. Z., 1991. Harmonious working and CSCW: computer technology and air traffic control. In *Studies in Computer supported cooperative work: theory, practice and design*, (ed.s), Bowers J.M & Benford S.D. North Holland: Amsterdam.

Hayes-Roth B. and Hayes-Roth F., (1979). A cognitive model of planning. *Cognitive Science*, 3, pp 275-310.

Hollnagel E. and Woods D.D., (1983). Cognitive systems engineering: new wine in new bottles. *International Journal of Man-Machine Studies*, 18, pp 583-600.

Hopkin V. D., 1971. Conflicting criteria in evaluating air traffic control systems. *Ergonomics* , 14, 5, pp 557-564.

Howes A. & Young R.M. 1997, The role of cognitive architecture in modeling the user: Soar's learning mechanism, *Human Computer Interaction*, Vol. 12 No. 4, pp 311-343

Hutchins E. (1994), *Cognition in the wild*. MIT Press: Mass.

Kanafani A., 1986. The analysis of hazards and the hazards of analysis: reflections on air traffic safety management. *Accident analysis and prevention*. 18, 5, pp403-416.

Keeney R.L., 1993, *Value focussed thinking: a path to creative decision making*. Cambridge MA: Harvard University Press.

Long J.B. and Dowell J., 1989. Conceptions of the Discipline of HCI: Craft, Applied Science, and Engineering . In Sutcliffe A. and Macaulay L. (ed.s). *People and Computers V.* Cambridge University Press, Cambridge.
Marr D., (1982). *Vision*. Wh Freeman and Co: New York.

Norman D.A., (1986). Cognitive engineering. In Norman D.A. and Draper S.W., (ed.s)*User Centred System Design*. Erlbaum: Hillsdale, NJ. pp 31 - 61.

Payne S.J., 1991, Interface problems and interface resources. In Carroll J.M (ed.) *Designing Interaction*, Cambridge University Press: Cambidge.

Rasmussen J., (1986). *Information processing and human-machine interaction: an approach to cognitive engineering*. North Holland: New York.

Ratcliffe S., 1985. Automation in air traffic management. *S. Journal of Navigation*, 38, 3, pp 405-412.

Rouse W. B., (1980). *Systems engineering models of human machine interaction*. Elsevier: North Holland.

Shepard T., Dean T., Powley W. and Akl Y., (1991). A conflict prediction algorithm using intent information. In *Proceedings of the Air Traffic Control Association Annual Conference*, 1991.

Sommerville, I., 1996, *Software Engineering*. Addison Wesley: New York.

Sperandio J. C., 1978. The regulation of working methods as a function of workload among air traffic controllers. *Ergonomics*, 21, 3, pp 195-202.

Vera A.H. and Simon H.A., 1993, Situated action: a symbolic interpretation. *Cognitive Science*, 17, pp 7-48.

Whitfield D. and Jackson A., (1982). The air traffic controller's picture as an example of a mental model. In *Proceedings of the IFAC conference on analysis, design and evaluation of man-machine systems*. Baden-Baden, Germany, 1982. HMSO: London.

Woods D.D. and Roth E.M., (1988). Cognitive systems engineering. In Helander M. (ed.) *Handbook of Human Computer Interaction*. Elsevier: North-Holland.

Appendix 1. Functions for computing ATMQ (fl): the air traffic management qualities for completed flights.

Quality of Safety Management for completed flights:

$$QSM = \sum(\text{QSM values for each event})$$

where the value of QSM for each event is given by the following rule:

```
if, time to collision = FALSE (i.e., no projected collision, then QSM = 0
else,
if time to collision is less than 5 minutes, then QSM = C (a penalty constant)
else, QSM = - (1 + (time to collision /projected total time of flight))
```

This rule means that if at a given airtraffic event, two aircraft are on a collision course and are less than a safe separation apart (300 seconds), then a penalty is immediately given, commensurate with a 'near miss' condition. When aircraft are on a collision course but a long way apart, safety is assessed as a function of closing time and projected time of complete flight. The form of function which this rule supplies is such that QSM is optimal when a value of zero is returned, meaning that at no time was the aircraft in separation conflict or on a course leading to a conflict no matter how far apart. The value increases negatively when conflict courses are instructed, and sharply so (as given by constant C) when those courses occur with less than a specified track and vertical separation.

Quality of flight progress management for completed flights:

$$QPM = \frac{\text{(goal total flight time) - (actual total flight time)}}{\left|\text{((goal total flight time) - (initially projected total time)+ const)}\right|}$$

Quality of fuel use management for completed flights:

$$QFM = \frac{\text{(goal total fuel use) - (actual total fuel use)}}{\left|\text{((goal total fuel use) - (initially projected total fuel use)+ const)}\right|}$$

Quality of exit variation management for completed flights:

$$QXM = - \frac{\left|\text{((goal exit height) - (actual exit height)+ const))}\right|}{\left|\text{((goal exit height) - (entry height))}\right|}$$

Quality of manoeuvres management for completed flights:

$$QMM = - \text{LOG10} \; \frac{\text{(actual number of manoeuvres)}}{\text{(goal number of manoeuvres)}}$$

The forms of function of the unit-less indices provided by these ratios are such that in each case, quality of management is optimal when a zero value is returned, that it to say, when actual state and goal state are coincident. QPM and QFM are greater than zero when respective actual states are better than goal states, and less than zero when they are worse (it is possible for actual values of fuel consumed or flight time to be less than their goal values). The difference is given by proportion with the difference that would have been the case if there had been no interventions by the ATM worksystem over the scenario. In this way, the added value of the worksystem's interventions is indicated.

The values of QXM increase negatively from zero with the difference between actual exit height and the goal exit height. The difference is again given by proportion with the difference that would have been the case if no ATM worksystem interventions had been made: the aircraft would have left the sector at its entry height.

The values of QMM range from +0.3 when the actual number of manoeuvres is less than the goal number of manoeuvres, to zero when actual and goal are equal, and slowly increase negatively as the number of manoeuvres increases above the goal number.

The constants in the formulae for QPM, QXM, and QFM are included to reduce the 'order effect' distortions when small differences occur in denominator or numerator. These constants are determined by numerical iteration to ensure a negligible change in the general shape of the functions.

Appendix 2. Functions for computing ATMQ (int): the air traffic management qualities for each controller intervention.

We can determine ATMQ(int) for any given time in the relationship between the previous state, the state following the intervention, and the desired state. For QPM, QFM and QXM, these states are final states projected over the remainder of the flight, and assume no further intervention will be made.

For each intervention, $\text{QEM} = \frac{\sum \text{QPM}}{n} + \frac{\sum \text{QFM}}{n} + \frac{\sum \text{QMM}}{n}$

where n = number of aircraft on the sector at the time of the intervention

Quality of flight progress management for each aircraft at each intervention

$$\text{QPM} = \frac{\text{(goal total flying time) - (new projected total flying time)}}{\text{((goal total flying time) - (previous projected total flying time)+ constant)}}$$

Quality of fuel use management for each aircraft at each intervention

$$\text{QFM} = \frac{\text{(goal total fuel use) - (new projected total fuel use)}}{\text{((goal total fuel use) - (previous projected total fuel use) + constant)}}$$

Quality of manoeuvres management for each aircraft) at each intervention

$$\text{QMM} = -\text{LOG10}\ \frac{\text{(current total of manoeuvres)}}{\text{(goal number of manoeuvres)}}$$

QXM is computed from the final event within a vector, since it is a closure-type task attribute. Safety is a continuous attribute, and QSM for each intervention is therefore as already computed for ATMQ(fl), as given in Appendix 1.