Informing the Specification of a Large-Scale Socio-technical System with Models of Human Activity

S. Jones, N.A.M. Maiden, S. Manning, J. Greenwood[1]

Centre for Human-Computer Interaction Design, City University, London
[1]National Air Traffic Services, London, UK
Contact e-mail: *s.v.jones@city.ac.uk*

Abstract. In this paper, we present our experience of using rich and detailed models of human activity in an existing socio-technical system in the domain of air traffic control to inform a use case-based specification of an enhanced future system, called DMAN. This work was carried out as part of a real project for Eurocontrol, the European Organisation for the Safety of Air Navigation. We describe, in outline, the kinds of models we used, and present some examples of the ways in which these models influenced the specification of use cases and requirements for the future system. We end with a discussion of lessons learnt.

Key words: use cases, specification, socio-technical systems, domain knowledge

1 Introduction

The literature in requirements engineering is replete with references to scenario- or use case-based approaches to requirements elicitation, specification and validation. However, much less is said about where the scenarios and use cases, which are the basis of such approaches, might come from. There is plenty of guidance on how, or in what style, use cases or scenarios should be written - see, for example, [1]. But what about the raw materials? How do we know what should go into a use case, or even what use cases to include in a specification in the first place?

Traditional approaches to systems analysis, such as SSADM, start by modeling the current system. This is done at a high level of abstraction, where models represent business events and rules, data and information flows. More recent approaches, such as Volere [2], recommend that one of the first steps in learning what people need should be to model the business which a new product will support, in order to obtain a first cut model of actors and use cases for the future system. Again, this is done at a high level of abstraction, where there is a great deal of similarity between the current and future systems. The Unified Process also states that actors and use cases for the initial use case model should be derived from high level business and domain models. For example, Arlow and Neustadt [3] recommend that use case modeling should begin by identifying actors and then considering how those actors will use the future system. They also provide a list of questions concerning storage and retrieval of information, and notification of external events and system state changes, which the analyst can use to help refine the list of use cases initially identified. The level of granularity at which initial specifications should be pitched is discussed by several authors - see, for example, [4] - but the main type of information about the current situation on which future specifications are to be based is usually information about actors and their goals. Finally, it is noticeable that much work on use case modeling tends to have a forward-looking focus on the 'vision' [5] or 'mission statement' [6] for the future system. This can be problematic in situations where there is a large and complex system already in place, and where a future system must be developed as an evolutionary step forward from a current system, rather than a revolutionary fresh start.

The discipline of human-computer interaction (HCI) provides a different perspective on the development of future systems. The HCI community has developed a different range of concepts for reasoning about socio-technical systems, which focuses more on the human components of such systems than is commonly the case in software engineering. There is also a strong tradition of using the results from in-depth studies of current work to inform the design of future systems. A small number of studies have been reported in the literature, in which rich and fine-grained observations of human behavior in existing systems have been used to inform the specification of future complex and large-scale socio-technical systems. For example, Viller and Sommerville [7] report the use of ethnographic studies to help identify use cases in a case study also based in the domain of air traffic control. Their approach, called Coherence, focuses on the impact of social analysis of existing systems on the design of future systems. Bisantz et al [8], [9] have reported studies investigating the utility of cognitive work analysis models in the design of large-scale socio-technical systems such as a next-generation US Navy surface combatant. In particular, Bisantz et al [9] point to pragmatic considerations which are important in selecting and adapting methods of cognitive work analysis to fit the demands of a time-pressured design situation, and point out the significance of developing work products which are timely and tightly coupled to other elements of the design process in this context.

In this paper we present our experiences and observations following an attempt to apply a range of HCI concepts and techniques to capture and record information about an existing socio-technical system in order to inform the development of a use case specification for a future system, called DMAN, in the domain of air traffic management. Our aim has been to develop a practical means by which requirements and systems engineers could use inputs from the HCI community to improve their practice in the specification of socio-technical systems. The data we present concerns a real project, carried out in a complex, safety-critical domain and within commercial constraints. Our approach to data analysis has therefore been mainly qualitative, rather than quantitative, since it was not possible to carry out controlled experimentation within these constraints.

Section 2 presents a brief overview of the DMAN project and the RESCUE requirements process, which provides the framework within which our work was carried out. Section 3 describes our choice of concepts for inclusion in models of human activity in the current system, and section 4 presents some more detailed observations regarding the way in which these concepts were used. We end with a discussion of lessons learnt and directions for future work.

2 DMAN and the RESCUE Process

The data presented in this paper relate to work carried out in the specification of operational requirements for DMAN, a socio-technical system for scheduling and managing the departure of aircraft from major European airports such as Heathrow and Charles de Gaulle. DMAN is a system that will support controllers in managing the process of departure from an airport and through the Terminal Manoeuvring Area (TMA). One DMAN system will manage all civil Instrument Flight Rules (IFR) departures from all airports within a TMA. DMAN will assist controllers in maintaining a high level of throughput while respecting all spacing constraints.

The specification for DMAN was developed by a requirements team which included engineers from UK and French air traffic service providers. These engineers modeled the DMAN system and requirements using techniques from the RESCUE requirements process. RESCUE - Requirements Engineering with Scenarios for a User-centred Environment – is a concurrent engineering approach, which allows us to integrate current HCI techniques and research perspectives with current best practice in relation to use-case based requirements specification. The RESCUE process has already been described in a number of other publications - see, for example, [10] - and in this paper, we provide just a brief overview.

RESCUE was initially developed to specify operational requirements for a system called CORA-2, a system that will provide computerised assistance to air traffic controllers to resolve potential conflicts between aircraft [11]. The RESCUE process has since been applied in the specification of requirements for DMAN, as described in this paper and in [12]; MSP, a system for scheduling aircraft from gate to gate across multiple, multi-national sectors [13]; EASM, a system to support enhanced airspace management [14]; and VANTAGE, a project aimed at minimizing the environmental impact of regional airports.

The RESCUE process was developed by academic researchers from the domains of HCI and requirements engineering, working with staff at Eurocontrol, the European Organisation for the Safety of Air Navigation, and was specifically targeted towards the needs of the domain of air traffic management. Thus RESCUE focuses on specification of requirements for critical systems, where development of new systems is evolutionary rather than revolutionary, and where the emphasis is on getting requirements right, rather than speed to market.

RESCUE is aimed at the specification of operational requirements – relatively high-level requirements which are typically concerned with the overall functionality of the socio-technical system, the division of labor between human and technical components of the system, and basic statements of non-functional requirements or constraints concerning usability, training, look and feel etc. Detailed specification of presentation in the user interface, user interaction and information architecture comes at a later stage in the development lifecycle.

The CORA-2, DMAN and MSP projects in which RESCUE has been applied are part of the European Air Traffic Management's Automated Support to Air Traffic Services (ASA) programme, whose aim is to develop concepts, requirements and procedures for the provision of tools to enhance the air traffic control decision-making process. The ASA programme as a whole has adopted the principle of 'human-centred automation'. This principle asserts that 'the human bears the ultimate responsibility for the safety of the aviation system', and that the controller must therefore remain in command of the system. The system, in turn, must provide information consistent with controllers' responsibilities, and presented in a format meaningful to controllers in a given context so that controllers can monitor and understand what their automated systems are doing. Proper consideration of the human element in the system therefore had to be included in our process.

The RESCUE process consists of a number of sub-processes, organised into 4 on-going streams. These streams run in parallel throughout the requirements specification stage of a project, and are mutually supportive. The four RESCUE streams focus on the areas of:

- Analysis of the current work domain using **human activity modeling** - this stream will be described in more detail below;
- **System goal modeling** using the *i** goal modeling approach;
- **Use case modeling** and specification, followed by systematic **scenario walkthroughs** and scenario-driven **impact analyses**;
- **Requirements management** using VOLERE [2] implemented in Rational's requirements management tool RequisitePro in current rollouts of RESCUE.

In addition to these four streams, the RESCUE process uses the **ACRE** framework to select techniques for requirements acquisition, and **creativity workshops,** based on models of creative and innovative design, to discover candidate designs for the future system, and to analyse these designs for fit with the future system's requirements.

This paper builds on work described in [15] and focuses on the relationship between the human activity modeling and use case modeling streams.

3 Concepts Used in Models of Human Activity for DMAN

Human activity modeling in RESCUE focuses on the activity of humans in the current system. This is in line with the principle of human-centred automation defined above. In the human activity modeling stream of the RESCUE process, the project team needed to understand and model the controllers' current work in order to facilitate the specification of technical systems that could better support that work.

The human activity modeling stream in RESCUE consists of two sub-processes – data gathering and human activity modeling. During the first sub-process, data about all components of the activity model are gathered and recorded, initially in a relatively unstructured way. Techniques to gather this data are familiar to those in the domains of both HCI and RE and include: observation of current system use; informal scenario walkthroughs, using scenarios that describe how the current system is used;

interviews with representative human users; and analysis of verbal protocols, or recordings of users talking through scenarios or tasks.

In the second sub-process, the project team creates a *human activity model* by generating a number of *human activity descriptions* corresponding to each of the major types of activity in the current system. This is analogous to the creation of a use case model for the current system, consisting of a number of related use case descriptions (UCDs), although the kinds of information recorded in human activity descriptions are different from those which would be included in use case descriptions, as described below. Once created, the human activity model is used to inform the development of use case descriptions for the future system during stage 2 of the RESCUE process. It is also used to validate the completed use case descriptions. The rest of this section explains what kind of human activity model was used in DMAN and why.

3.1 Basic Concepts

The categories of concepts for use in human activity descriptions in the DMAN project were chosen with reference to the literature of task analysis, cognitive task analysis and cognitive work analysis as explained in [15]. In summary, concepts used in DMAN human activity models were as follows:

- Human actors - people involved in system;
- Goals: states of the system which one or more actors wish to bring about – where goals may be
  - high-level functional goals relating to the system as a whole, or local goals relating to particular tasks;
  - individual goals, relating to single actors, or collective goals, relating to teams of actors;
  - prescribed goals or non-prescribed goals
- Actions: undertaken by actors to solve problems or achieve goals – where higher level, generic actions may be broken down into component physical, cognitive or communication actions
- Resources: means that are available to actors to achieve their goals;
- Resource management strategies: how actors achieve their goals with the resources available;
- Contextual features: situational factors that influence decision-making,; and
- Constraints: environmental properties that affect decisions.

3.2 Additional Concepts for Structuring Models of Human Activity

After deciding what concepts to include in our model of human activity, our next question concerned the way in which information relating to each of these concepts should be structured in order to provide useful inputs into the use case writing process. We decided to model activity in terms of a script-like representation, as the majority of the knowledge to be modeled was procedural, concerning the sequences of actions which take place under various circumstances, and we also felt that this would map easily onto the script-like use case specifications of the future system which were our final target. We designed a template, within which we could record knowledge relating to each of the concepts identified above in a script-like format which would also provide space to record:

- Administrative information, including the author, date and source of information included, thus enabling traceability;
- A brief précis of the content of the human activity description, analogous to the kind of précis commonly included in use case descriptions;
- A triggering event, suggested by our consideration of scripts above;
- Any pre-conditions which are necessary for the activity to take place – again this was included because pre-conditions are normally included in use case descriptions, and
- Differences due to variations – different but normal or equally valid ways of achieving the relevant goal(s), as suggested by our consideration of scripts, and again as typically included in use case descriptions.

One completed template is referred to as a Human Activity Description (HAD), and a Human Activity Model (HAM) consists of a number of HADs, as stated earlier. An example showing extracts from a completed HAD template is shown in figure 1, where we can see how knowledge about each of the concepts identified above can be placed within such a script-like representation.

Figure 1 shows extracts from one of the HADs developed for the DMAN project. It describes what happens when a pilot calls one of the air traffic controllers, the Ground Movement Controller (GMC), to request clearance to push back, or leave the stand ready for take-off. Different parts of the description relate to the activity as a whole or to particular actions, thus providing a structured but flexible description of current work practices. For example, actors, goals, contextual features and constraints relate to the activity as a whole, while different resources and resource management strategies may relate to different actions. Note also that actions in the normal course of the human activity description are broken down into their physical, cognitive, and communicative components.

Figure 2 shows extracts from a use case description developed for DMAN, in which we can see the similarities between the concepts and structures used in the human activity description, and those in the use case description. For example, there are fields for describing actors, précis, triggering events, pre-conditions, normal course (i.e. a sequence of actions), and variations in both HADs and UCDs. The relationships between some of the remaining concepts will be explained in the following section.

4 Human Activity and Use Case Modeling in DMAN

In this section, we provide further information about the generation of human activity models for DMAN and the relationship of these models with DMAN use cases.

HAD10	**Runway ATCo Gives Line Up Clearance**
Author	……………..
Date	……………..
Source	ATC meeting 6th March / 2nd April 2003
Actors	Runway ATCo, Pilot
Precis	To decide when the next aircraft should line up and to communicate line up clearance to the pilot.
Goals	Decision made as to when the next aircraft can line up Pilot given line up clearance Strip positioned correctly in the bay LVP or MDI procedures adhered to, if in effect
…	…
Triggering event	Previous aircraft has received clearance to take off OR Runway ATCo decides that line up is appropriate
Preconditions	Aircraft at holding point
Normal course	1. Departure/Air controller decides which aircraft can next line up and when Resources – strip Physical actions – touch strip, look at airfield, aircraft, holding point and runway, move to look out of window Cognitive actions – read strip information, validate visually, recognise aircraft and match with strip, recognise when it is appropriate to give line-up clearance, formulate aircraft line up clearance sequence, understand current airspace, runway and capacity situation
	2. Runway ATCo calls Pilot and gives line up clearance Resources – strip, radio, headset Physical actions – touch strip, flick radio transmission switch, look at aircraft, runway and holding point, move to look out of window Communication actions – talk to pilot, issue clearance, provide information Cognitive actions – read strip information, validate visually
	3. Pilot confirms details etc
Differences due to variations	….
Contextual features	1. If the aircraft has a problem, i.e. technical delay, technical failure or emergency, the pilot may call the controller Resources – strip, radio, headset Physical actions – touch strip, flick radio transmission switch, look at aircraft, runway and holding point, etc ………………
Constraints	Bay size – limited space for strips Noise levels – printer, system alarms, people talking Staff shortage ……………..

Fig. 1. Extracts from a Human Activity Description for DMAN

4.1 Data Collection and Generation of Human Activity Models

For DMAN, the data to be used in building the Human Activity Model was collected during the course of 2 half day visits to the Visual Control Room (the control tower) at Heathrow, during which controllers were observed at work, and subsequently interviewed. An informal scenario walkthrough session was held about 2 weeks later, with air traffic controllers from Heathrow and Gatwick. The major effort of producing the HAM involved one full-time worker for approximately 6 weeks. The human activity model for DMAN consisted of 15 separate human activity descriptions. Table 1 presents an overview of the numbers of elements of significant concept types identified in sections 3.1 and 3.2 in the HAM as a whole, and on average per HAD.

UC7	**Give Line Up Clearance**
Author	…………….
Date	…………….
Source	RESCUE stage 1 document
Actors	Runway ATCo, Pilot, DMAN, Departure clearance ATCo, A-SMGCS, TACT, FDPS, CDM system
Problem statement (now)	Integrate departure clearance into departure planning process
Precis	The runway becomes available for a new aircraft, or the Runway ATCo has a new aircraft under his/her control at a runway holding point. The runway ATCo selects the next aircraft to line up, optionally taking guidance from the DMAN recommended sequence. The Runway ATCo clears the pilot to line up for departure. The aircraft lines up. A-SMGCS records the aircraft's movement and sends an update of the aircraft status to DMAN.
Requirements	…
Constraints	…
Added value	
Justification	
Triggering event	The runway becomes available for a new departing aircraft.
Preconditions	The next aircraft in the DMAN departure sequence is under the control of the Runway ATCo
Successful end states	Aircraft receives line up clearance if appropriate Aircraft does not receive line up clearance if not appropriate
Unsuccessful end states	Aircraft receives line up clearance when not appropriate Aircraft does not receive line up clearance when appropriate
Normal course	1. The Runway ATCo looks at the DMAN recommended sequence
	2. The Runway ATCo looks at the aircraft holding by the runway
	3. The Runway ATCo decides that the next aircraft in the DMAN sequence can line up on the runway
	etc
Variation 3	IF the entire runway is clear and the required separation from the previous aircraft has elapsed THEN replace step 3 with:
	3a The Runway ATCo clears the pilot to line up and take off

Fig. 2. Extracts from a Use Case Description for DMAN

Table 1. Overview of concept distributions

Concept	Total no. in HAM	Avg. no. per HAD	Range across HADs
Actor	45	3	1 – 7
Goal	76	5	2 – 11
Triggering event	18	1	1 – 2
Precondition	19	1	0 – 3
Action (generic)	127	8	5 – 14
• physical action	221	15	4 – 24
• communication action	99	7	0 – 16
• cognitive action	255	17	3 – 29
Resource	201	13	5 – 27
Resource management strategy	25	2	0 – 4
Differences due to variations	38	3	0 - 5
Contextual features	74	5	1 – 9
Constraints	136	9	4 - 11

4.2 Usefulness of HAD Concepts

The completed HADs were made available to the engineer responsible for writing the DMAN use case descriptions. Note that the engineer also had access to other sources of information developed as part of the RESCUE process, including a rich context model, a use case diagram, and ideas generated in the course of a 2 day creativity workshop - see [10], for further information. After writing the use case descriptions, this engineer was asked to provide feedback on the utility of the HADs, and particular concepts represented within them through a questionnaire. In the questionnaire, each of the concepts was rated for usefulness on a scale of 1 – 5, where 1 meant 'HADs were not useful at all in writing UCDs – it would have made no difference whether they were available or not', 3 meant 'HADs were quite useful in writing UCDs' and 5 meant 'HADs were essential in writing UCDs – I couldn't have done it without them'. There was also space for providing more general comments. Overall, HADs were judged, by the engineer who wrote the use case descriptions, to be most useful in writing UCDs involving sequences of prescribed behaviors, for example in interactions between pilots and controllers. Table 2 shows the relative usefulness ratings for individual concepts within the HADs.

In addition to their use in writing use case descriptions, human activity descriptions also played a significant role in validating first draft use case descriptions. Using the human activity model, a total of 23 issues were identified for discussion in relation to the first draft use case specification. Feedback provided by members of the requirements team on this basis was judged by the original author of the use case descriptions to be 'very useful'. We return to this issue below.

Table 2. Overview of concept utility

Concept	Usefulness rating
Actor	4
Communication action	4
Action (generic)	3
Cognitive action	3
Differences due to variations	3
Triggering event	2
Precondition	2
Physical action	2
Resource	2
Goal	1
Resource management strategy	1
Contextual features	1
Constraints	1

The results we present in the rest of this section are based on a qualitative exploration of the data arising from the DMAN project. It would not be meaningful to attempt a precise quantification of the extent to which constructs in the human activity model relate to those in the use case model. This is because while some elements of the human activity descriptions can be imported directly into the use case descriptions, others exert a more subtle influence, or appear in modified form, as will be seen below. In the following paragraphs, we attempt to give a flavor of the relationships between HADs and UCDs as a whole, and then between individual constructs in the human activity and use case descriptions.

4.3 Overview of Relationships Between HADs and UCDs

The strength of relationships between HADs and UCDs was estimated by considering the similarity of constituent actors, précis, actions, triggering events, goals/end states and variations. On this basis, 11 out of the 15 HADs were judged to have some relationship with UCDs in the future system specification. HADs 5, 6 and 10 had strong relationships with UCDs 3, 4 and 7 respectively, and there was a lot of similarity between the sequences of actions described in each case. On the other hand, much of the human activity, especially the cognitive actions, described in HAD8 ('departure/air controller calculates departure sequence') and HAD9 ('optimisation sequence') was to be taken over by the DMAN system, so the relationship between these HADs and the relevant UCDs was more complex, as will be discussed below. HAD1 ('receive and prepare flight strip'), HAD11 ('departure/air controller gives take off clearance'), HAD 12 ('flight strip logging') and HAD 13 ('SVFR clearance procedure for aircraft') do not correspond directly to any UCDs as these are activities in which DMAN will not play any role. Each of the remaining HADs, was weakly associated with a UCD for the future system.

4.4 Use of Individual Concepts from the Human Activity Descriptions

In this section, we present examples to illustrate the kinds of relationships which existed between concepts in the HAM and those in the future system specification.

Actors. Actors were judged to be very useful in writing use case descriptions. They were typically carried over into the relevant UCDs, with some renaming of actors - the Ground Movement Controller became the Ground Air Traffic Controller to reflect some changes in responsibilities - and some new actors, such as the A-SGMS ground radar system, being added in the future system.

Goals. As stated above, goals were intended to be states of the system which one or more actors wish to bring about. HAD goals were recorded at various levels of abstraction, some relating to high-level functional requirements for the future system, and some relating to particular actions. Most of the goals identified were collective goals, relating to the system as a whole. Only 2 out of a total of 76 related more to individual workers. These concerned the desire to regulate workload, for example 'Runway ATCO workload regulated.' (HAD7). Only one of the goals in the Human Activity Model which was delivered to the customer was a non-prescribed goal ('Aircraft adhered to targets on meeting the estimated push back time.' – HAD4).

Goals were rated by the engineer as ' not useful' in writing use case descriptions. However, on analyzing the future system specification, it was found that goals in the HAM typically translated either into successful end states in the relevant UCD, or directly into requirements. For example, the goal 'Pilot given taxi clearance' (HAD6) is expanded into two successful end states for UC4: 'Aircraft is cleared to runway holding point' and 'Aircraft is cleared to intermediate point on the taxi route'. The goal 'Slot time adhered to' (HAD6) is operationalised in the requirement 'FR2: DMAN shall support ATCO to respect CFMU slots' and the goal 'Timely taxi clearance given' appears in the specification of the future system as the performance requirement 'PR12: ATCO using DMAN shall give timely taxi clearance'.

Triggering events. For the 11 HADs with relationships to particular UCDs, 5 of the triggering events mapped onto similar triggering events in the relevant UCDs. For example 'Pilot calls for start up' (HAD3) appears as 'Pilot requests start up clearance' in UCD2. 3 of the triggering events from these HADs were expanded to significantly more complex conditions in the relevant UCDs. For example 'Pilot calls for taxi' (HAD6) is expanded to 'Pilot requests taxi clearance OR taxi route becomes clear of other conflicting traffic OR all aircraft planned for departure in advance of this one are now ahead on the taxiway' in UCD4.

Preconditions. Once again considering the 11 HADs with relationships to particular UCDs, only 2 of the pre-conditions identified in HADs mapped onto similar pre-conditions in UCDs. For example 'Pilot is ready to start' (HAD4) appears as 'Flight cleared for start up' in UD3. Other pre-conditions listed in the UCDs are much more concerned with specifying relevant states of the DMAN system.

Actions. Generic actions were specified at a similar level of abstraction to those in the normal course of a use case description, for example: 'Pilot calls for taxi' (action 1, HAD6). Then, the set of lower level physical, communication and cognitive actions done by the human actor, usually an air traffic controller, in association with the generic action were recorded, as shown in figure 1. There was a wide variation in the number of lower level actions recorded for a single generic action. Some generic actions had no lower level actions associated with them. This was often the case where the generic action was performed by an actor other than an air traffic controller. Others had up to 8 lower level, especially cognitive or physical actions associated with them.

Actions, especially communication actions, were judged by the engineer who wrote the use case descriptions to be very useful. They were particularly helpful in writing use cases where the introduction of DMAN did not change the course of events, for example where pilots and controllers must continue to interact in a prescribed fashion. Some of the generic actions mapped directly onto UCD actions, for example: 'Pilot calls for taxi' (action 1, HAD6) mapped to 'The pilot requests taxi from the Ground ATCO' (action 1, UCD4). Some mapped onto a version of the action in which DMAN is providing support. For example: 'GMC locates strip in bay' (action 2, HAD6) mapped to 'The Ground ATCO [GMC] finds the flight in the DMAN planned departure sequence.' (action 2, UCD4).

Often, however, the relationship between actions in the current system and those to be carried out in the future system was more complex. The goals of DMAN, as described above, were basically to support controllers in achieving maximum TMA and runway capacity, without increasing their workload, or in other words, to increase the numbers of aircraft controllers are able to manage by reducing the amount of effort required per aircraft. One obvious approach to this was to reduce the amount of cognitive effort required in order to manage aircraft departures. Thus DMAN was designed to support some of the more difficult cognitive tasks, such as formulating an aircraft line up clearance sequence, and co-ordinating inbound taxiing aircraft, towed aircraft, aircraft crossing the runway and other taxiing aircraft with aircraft departures, by calculating a proposed departure sequence which controllers could adopt and use, if they judged it appropriate, rather than requiring controllers to formulate such a sequence themselves as a purely cognitive activity without support. An example of this can be seen in the relationship between HAD10 and UCD7, as shown in figures 1 and 2, where perhaps the most difficult cognitive activity - 'Formulate aircraft line up clearance sequence' (part of action 1, HAD10) - has been taken over by DMAN, as reflected in the requirements FR68: DMAN shall calculate the departure sequence' and 'FR69: DMAN shall provide ATCO with departure sequence information', while the human controller still has ultimate control over decisions made, and is still required to carry out visual checks (action 2, UC7), shown as physical actions (part of action 1, HAD10) in the Human Activity Model, before acting on DMAN's advice.

Finally, it should be noted that the detailed information contained in the HAD actions was particularly useful in validating first draft use case descriptions. Of the 23 issues identified for discussion as part of the validation exercise, 13 related to actions in the HAM.

Resources. The same resources were often referred to at different points within HADs, and within the Model as a whole. Only 26 different resources were identified as being relevant anywhere in the system. Resources were not judged to be very useful in writing use case descriptions, as they would be significantly different under DMAN. For example, paper flight strips would be replaced by electronic flight strips once DMAN was introduced.

Resource management strategies. Resource management strategies (RMS) were very infrequently identified. RMS were only identified as relevant for 25 actions in the Human Activity Model as a whole, with an average of 2 per HAD. Only 2 different RMS were identified in the Model as a whole. Resource management strategies were judged as 'not useful at all' (rating 1 our of 5) in writing Use Case Descriptions.
Differences due to variations. Different practices by different controllers, and in different airports were recorded in this section of the HAD template. A total of 38 different variations were recorded in the Model as a whole. In some Descriptions, no variations were identified, whereas in others, there were up to 8. This field in the Description template was rated 'quite useful' (3 out of 5) in writing Use Case Descriptions, as it gave information on the different, but equally valid, ways of carrying out relevant tasks which may need to be supported in the future DMAN system. Some examples of variations identified in the HAM were: 'Ground Movement Controller may aid optimal sequencing' - not all GMCs do this; 'For Gatwick, remote holds are offered to aircraft', which is different from other airports; 'For inbound aircraft, the aircraft reaches the stand', where the normal course in the HAD refers to outbound aircraft.

As an example, the first of these lead to the identification in UC4 of a variation: 'If the aircraft requested taxi previously but clearance was refused because of taxiway congestion, then replace step 1 with 1a: The Ground ATCO sees that a requested taxiway is now free of congestion'.

Contextual features. This section of the HAD template was intended to be used to record what happens under unusual or irregular circumstances. For example in HAD1 we have: 'If there is an airport, airfield or airspace emergency situation i.e. fire, bomb alert, etc, then activity may be stopped'. Contextual features were judged 'not useful at all' (rating 1 out of 5) in writing Use Case Descriptions. However, they were used to identify different possible contexts for scenario walkthroughs, which in turn helped to identify requirements specifying how the future system should work in exceptional circumstances. For example, the requirement: 'FR26: DMAN shall provide a bad weather/emergency incident option' was identified in the scenario walkthrough for use case 1 and lead to the identification, through decomposition, of 5 additional functional requirements (FR27 – 31) concerned with the provision of an emergency incident option in DMAN.

Constraints. Almost all of the constraints identified were the same for each HAD. Most of the constraints identified related to the physical environment in which controllers operate. However one constraint, described as 'staff shortage' identified a number of times related more to the organisational environment. Constraints were judged as 'not useful at all' (rating 1 out of 5) in writing Use Case Descriptions. However, they did have implications for system requirements. For example, the constraint of staff shortage, is reflected in requirements: 'UR10: DMAN shall not increase workload in order to display sequence info' and 'FR81: DMAN shall not replace the Ground ATCO or Departure ATCO, but aid them in workload' and in the rationale to many other requirements where it is acknowledged that workload must not increase.

5 Discussion

In this paper we have presented our experience and observations of work carried out in the DMAN project. In this case, we were dealing with the specification of high-level operational requirements, for a critical system, where development of the new system would be evolutionary rather than revolutionary, and where the emphasis was on getting requirements right, rather than speed to market. There was also a need to follow the principle of 'human-centred automation', which meant that proper consideration of the human element in the system had to be included in our process. We therefore developed a template for Human Activity Descriptions, which allowed us to build a richer and more detailed model of the current system than is typically used in use case-based system specification. We aimed to build on the work of both Viller and Sommerville [7] and Bisantz et al [8], [9] to develop a practical approach to the explicit recording of knowledge about the existing socio-technical system which would enable systems engineers to develop and critique a use case specification of the future system.

In summary, our observations regarding the benefits of our approach in this project, as presented in section 4.4, are as follows:

- Descriptions of cognitive actions in the human activity model were particularly useful in identifying points where the controller needed additional support from the new DMAN system.
- Descriptions of communication actions were judged to be very useful in writing use case descriptions, as these would remain unchanged in the future system. However, many physical actions, such as 'touch strip', were simply artefacts of the way in which the current system worked, and so were not relevant in the future system.
- Variations in the human activity model were useful and mapped directly into variations in the use case model, as did triggering events.
- Constraints in the human activity model gave requirements for the future system.
- Contextual features in the human activity model gave contexts for scenario walkthroughs.

Of course, these benefits come at a considerable cost in terms of the effort required to generate the human activity model, and our approach would not be suitable in every context. In order to retain benefits such as the above, while minimizing the costs, we intend in our next project to use a more iterative approach to the development of the human activity model. We will begin by developing a human activity diagram, analogous to a use case diagram, and will use this to focus a second stage of human activity modeling efforts on those parts of the current system which will be most affected by the proposed future system. For example, in the case of DMAN, it would have been helpful to have more detail in the human activity model about the way in which controllers handle their strips - a part of their work which will be strongly influenced by the introduction of DMAN, and less on how they use the radio - a part of the current socio-technical system which will not be greatly affected by the introduction of DMAN.

We also plan some minor changes to the human activity description template so that concepts such as resources and resource management strategies which prompted a lot of repetition through the course of a single description would be modeled at the level of the description as a whole, rather than at the level of individual actions. In the same way, some concepts, such as constraints, might be better modeled as relating to the current system as a whole, rather than to individual human activity descriptions.

This leads us to our final point: the need for multi-disciplinary requirements and design teams in which communication between members of the team with different backgrounds and differing levels of domain knowledge is facilitated by the use of explicit representations of knowledge about the system which all members of the team can comprehend. On the basis of our experience in DMAN, we believe that human activity models, comprising human activity descriptions written using the template presented in this paper can provide this kind of support. Our work aims specifically to provide a way of dovetailing a range of HCI concerns with current best practice in use case authoring. We are therefore optimistic that our work might provide a useful basis for increasing collaboration between those from backgrounds in

HCI and requirements or systems engineering in the specification of requirements for socio-technical systems.

Acknowledgements. The authors wish to thank all members of the DMAN project team for their participation and support throughout the project, and all those at NATS, Sofreavia and Eurocontrol who have contributed to the development of the RESCUE process as a whole.

References

1. Alexander, I.F., Maiden, N.A.M.: Scenarios, Stories, Use Cases. John Wiley and Sons (2004)
2. Robertson, S., Robertson, J.: Mastering the Requirements Process. Addison Wesley (2005)
3. Arlow, J., Neustadt, I.: UML and the Unified Process. Addison Wesley (2002)
4. Larman, C.: Applying UML and Patterns. 2nd edn. Prentice Hall (2001)
5. Bittner, K., Spence, I.: Use Case Modeling. Addison Wesley (2003)
6. Kulak, D., Guiney, E.: Use Cases. Addison Wesley (2000)
7. Viller, S., Sommerville, I.: Ethnographically Informed Analysis for Software Engineers. Int. J. Human-Computer Studies **53** (2000) 169 – 196
8. Bisantz, A., Ockerman, J.: Informing the Evaluation and Design of Technology in Intentional Work Environments through a Focus on Artifacts and Implicit Theories. Int. J. Human-Computer Studies **56** (2002) 247 – 265
9. Bisantz, A., Roth, E., Brickman, B., Gosbee, L.L., Hettinger, L. McKinney, J.: Integrating Cognitive Analyses in a Large-Scale System Design Process. Int. J. Human-Computer Studies **58** (2003) 177 – 206
10. Jones, S., Maiden, N.A.M.: RESCUE: An Integrated Method for Specifying Requirements for Complex Socio-Technical Systems. In: Mate, J.L., Silva, A. (eds.): Requirements Engineering for Sociotechnical Systems. Idea Group Inc. (2005)
11. Maiden, N.A.M., Jones, S., Flynn, M.: Innovative Requirements Engineering Applied to ATM. In Proc. ATM 2003, 5th USA/Europe R&D Seminar, Budapest, June 23 – 27 (2003)
12. Maiden, N.A.M., Jones, S., Manning, S., Greenwood, J., Renou, L.: Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study. In Persson, A., Stirna, J. (eds.): Advanced Information Systems Engineering. Proc. 16th Intl. Conf. CAiSE 2004, Lecture Notes in Computer Science, Vol. 3084. Springer-Verlag (2004) 368-383
13. Maiden, N.A.M., Robertson, S.: Integrating Creativity into Requirements Processes: Experiences with an Air Traffic Management System. In Proc 13th IEEE Intl. Requirements Engineering Conf. (RE05). IEEE CS Press. (2005)
14. Maiden, N.A.M., Ncube, C., Robertson, S.: Can Requirements be Creative? Experiences with an Enhanced Air Space Management System. To appear in Proc. 29th Intl. conf. on Software Engineering (ICSE07). IEEE CS Press. (2007)
15. Jones, S., Maiden, N.A.M., Manning, S., Greenwood, J.: Human Activity Modelling in the Specification of Operational Requirements: Work in Progress. In Bridging the Gaps II: Bridging the Gaps between Software Engineering and Human-Computer Interaction, Workshop W1L, 26th International Conference on Software Engineering (ICSE2004). IEE. (2004) 1 – 8