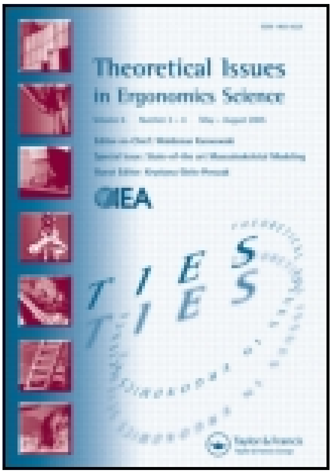

Theoretical Issues in Ergonomics Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ttie20

Specifying the requirements for requirements specification: the case for Work Domain and Worker Competencies Analyses

Rich Charles McIlroy [a] & Neville A. Stanton [a]

[a] Transportation Research Group, School of Civil Engineering and the Environment, University of Southampton , Highfield Campus, University Road, Southampton SO17 1BJ , UK

Published online: 08 Apr 2011.

Theoretical Issues in Ergonomics Science
Vol. 13, No. 4, July–August 2012, 450–471

Specifying the requirements for requirements specification: the case for Work Domain and Worker Competencies Analyses

Rich Charles McIlroy* and Neville A. Stanton

Transportation Research Group, School of Civil Engineering and the Environment, University of Southampton, Highfield Campus, University Road, Southampton SO17 1BJ, UK

(*Received 5 August 2010; final version received 2 November 2010*)

This article presents an argument for the applicability of Cognitive Work Analysis (CWA), particularly Work Domain and Worker Competencies Analyses, in supporting the requirements specification process for the acquisition of bespoke, socio-technical systems. We argue that the outputs of CWA should be included within requirements specification documents as they provide a comprehensive system model, in terms of constraints, opportunities and functional relationships, that would not be possible to represent in the current style of text-based requirements documents, and that communicating required system architectures to system designers is more effective using the graphical-based representations of CWA than by text alone. We also argue that the collaborative and iterative process of conducting a CWA should be continually performed throughout the acquisition cycle, involving Human Factors specialists, prospective end-users and subject matter experts.

Keywords: cognitive work analysis; requirements engineering; procurement; acquisition; system specification documents

1. Introduction

Requirements engineering can be considered as the branch of systems engineering that is concerned with the goals and functions of, and constraints acting upon systems, and the relationships between those factors (Nuseibeh and Easterbrook 2000); where *system requirements* describe the needs of the system and the environment within which that system will be situated, *requirements engineering* is the process by which these requirements are determined (Cheng and Atlee 2007). Requirements engineering is undertaken at the start of a system's lifecycle and, through the description of precise specifications for system behaviour and architecture, lays the foundation for the development of that system. In doing so, the resultant requirements specifications 'provide the basis for *analysing* requirements, *validating* that they are indeed what stakeholders want, *defining* what designers have to build, and *verifying* that they have done so correctly upon delivery' (Nuseibeh and Easterbrook 2000, p. 37). Although requirements specifications aim to document the needs of a system under development such that the envisioned system will successfully perform required functions and support user behaviour, for a variety of reasons this aim is not always met (Hsia *et al.* 1993). As the examples presented in Table 1 show, the cost of failing to successfully specify system requirements, and indeed failing to

*Corresponding author. Email: R.McIlroy@soton.ac.uk

ISSN 1463–922X print/ISSN 1464–536X online

http://dx.doi.org/10.1080/1463922X.2010.539287
http://www.tandfonline.com

Table 1. Large scale projects failures in the UK.

System	Taurus computerised share trading system for the London stock exchange	Channel tunnel	Computerised dispatch system for the London ambulance service	National (UK) air traffic control system	National health service (NHS) computer system
Approximate effort (years)	3	6	1	< 10	10
Approximate cost	£75 m	£4.6bn	£1.5 m	£339 m	£12bn
Aspiration[a]	'Paperless and computerised shareholdings . . . reduce time . . . bigger and better'	'One of the seven wonders of the modern world'	'More efficient . . . automation . . . greater capacity'	Existing system had 'functional limitations that would compel any modern engineer into laughter . . . '	'A grandiose IT project . . . transform the NHS'
Outcome	Scrapped	80% cost overrun	Scrapped[b]	Delayed and over-budget	Over-budget and reduced capability

Notes: [a]Aspiration derived from the BBC News website (www.bbc.co.uk/news).
[b]This failure was implicated in the loss of two lives.
Source: Adapted from Walker *et al*. (2009).

successfully interpret and implement those requirements, can have serious ramifications (Walker *et al*. 2009).

It is our contention that Cognitive Work Analysis (CWA; Rasmussen *et al*. 1994, Vicente 1999, Jenkins *et al*. 2009) can serve as a tool to support and facilitate the requirements specification process in such a way as to increase the chance for successful production, and subsequent interpretation and implementation, of system requirements, and in turn reduce the likelihood of acquisition failure. Whilst the Work Domain Analysis (WDA) phase of CWA has in the past been linked with requirements specification (e.g. Naikar and Sanderson 1999, Leveson 2000, Naikar 2006) and system procurement (e.g. Naikar and Sanderson 2001), a discussion of its relation to current processes, and of the other phases of CWA in relation to requirements specification, is currently lacking in the literature.

The aim of this article is, therefore, twofold; first, to provide links between the requirements specification process and CWA; second, to describe which CWA sections can be successfully applied, and how that application should proceed, in order to specify requirements that ultimately lead to usable, functionally relevant, affordable systems that do not incur considerable over-spend in post-deployment re-design and that do not necessitate extensive training schedules (see Stanton *et al*. 2009 for an example of such a system). It is the contention of the authors that only through the correct and enduring application of suitable Human Factors methods, CWA in particular, is it possible to minimise the risk of sub-optimal requirements specification and system design. This is a critical issue, especially for organisations that commonly procure bespoke systems (most government agencies, for example), as poor requirements specification, and hence poor system design, will likely result in systems that do not fully support user behaviour. Though the system described in Stanton *et al*. (2009) suffers from a plethora of issues relating to usability and functionality, it is argued that these problems could have been avoided had Human Factors principles been applied throughout the requirements specification and design stages of the system's lifecycle.

2. Linking requirements and CWA

Whilst the presentations of CWA and requirements documents differ significantly (CWA is, for the most part, graphical, whilst requirements documents are generally textual, sometimes presented in a tabular format), the process through which they are constructed bear some important similarities. Most requirements engineers agree that at the outset of any requirements specification process the domain or problem space needs to be defined; it is necessary to define the capability gap, and the domain in which that gap resides, before specifying any systems aiming to fill that gap (Jackson and Zave 1993). This is a central tenet of the first stage of CWA; WDA. In this stage of the analysis an Abstraction Hierarchy (AH) is constructed that describes the system across five levels of abstraction (Rasmussen *et al*. 1994, Vicente 1999, Jenkins *et al*. 2009). The description is based not on activities or goals specific to actors, like some other Human Factors methods (e.g. Cognitive Systems Engineering, Hollnagel and Woods 1999, Hierarchical Task Analysis, Stanton 2006, Salmon *et al*. 2010a or Cognitive Task Analysis, Schraagen *et al*. 2000), rather it is focussed on abstract functions and domain structure. The analysis does not focus on how activities are to be achieved, but on what needs to be fulfilled within the constraints of the domain. This concept of describing *what* needs to achieved, independent of *how* it is to be achieved, is central to the requirements documents used across organisations and industries. Both CWA (Rasmussen *et al*. 1994) and requirements specification processes are, or at least should be, problem or domain driven, independent of technology (in requirements engineering, this concept is akin to describing 'what' and not 'how'; e.g. Borgida *et al*. 1985). The concept that domain understanding must come before consideration of specific, technologically focussed activity is elegantly described by Ernst *et al*. (2006) in the context of software requirements engineering:

> This is akin to understanding the terrain before understanding what paths one can take therein. (p. 3)

With both requirements engineering and CWA comes the need for prior acquisition of information. It is not possible to analyse or describe any system without understanding the domain in which it is to be situated. It is crucial that a solid foundation of information regarding the domain and the actors is laid before specifying any system to be used in that domain, and by those actors. There are a number of requirements frameworks that make this point explicit (e.g. the KAOS methodology, Dardenne *et al*. 1991 and the i* framework, Yu 1997). Indeed, in a review of requirements engineering programmes in software development, Curtis *et al*. (1988) asserted that a poor understanding of the domain was the primary cause of project failure.

2.1. *Requirements and the AH*

There are a number of parallels that can be drawn between the differing levels of specificity expressed in CWA's AH and across requirements documents. For example, the User Requirements Document (URD) and the System Requirements Document (SRD) are the two most commonly constructed requirements documents in the UK Ministry of Defence's (MoD) requirements and acceptance process, whilst the US Department of Defence (DoD) uses Operational Requirements Documents (ORDs) and SRDs. For commercial software companies, the Software Requirements Specification (SRS) is often the only document produced. Though the discussion hereon in is in reference to the UK MoD's documents, they are close enough in nature to the requirements documents used by the US DoD (the ORD being analogous to the URD and the two SRDs being analogous) that the descriptions and resulting arguments put forward in the following sections may be considered as addressing both organisations. Furthermore, although SRS is commonly the only document produced (van Vliet 2000), its structure bears significant resemblance to the multiple document approach adopted by the MoD and the DoD.

For the purposes of comparing CWA with the requirements process, only brief descriptions of the URD and SRD are required; for a more detailed description of the contents and style of these documents the reader is referred to the UK MoD's Acquisition Operating Framework website (AOF 2010a).

First, the URD provides a description of the 'gap' between an existing capability and a required capability in terms of the outcomes, effects and services that a user needs to achieve or deliver. They are described in relation to the environment in which the system will be deployed, and are expressed in a way that is technology independent.

Second, the SRD offers a description of the requirements of the system that are traceable back to the requirements defined in the URD. They are more specific than those set out in the URD and supply a technical view of the functionality required by the system in order to fulfil user requirements and deliver the capability.

The effect of moving from more abstract to more specific requirements as the process progresses from URD to SRD, and indeed from ORD to SRD (in the case of the DoD) and across sections of the SRS (in the case of commercial software systems), is inherently similar to the style of the AH in CWA. At the top layer of the AH, the description of the system is highly abstracted; it provides a description of the overall purpose of the system without describing any of the functions or objects involved in achieving that purpose. For each level down, the descriptions become more detailed until the lowest layer, in which the physical objects comprising the system are described. Although the SRDs and the latter sections of the SRS do not descend to this level of granularity, it is in the process of travelling from abstract to specific, retaining the connections all the way (functional connections in the AH and traceability connections in requirements documents) that the similarities present themselves. Furthermore, both the SRS and SRDs are designed to present a functionally organised hierarchy of levels of detail, where moving up a level describes what is being supported (i.e. why) and looking down a level describes its realisation (i.e. how), therefore mirroring the means–ends causal chains that characterise the AH (Vicente 1999; Figure 1).

In addition to the progression through levels of abstraction, further links can be made between CWA's AH and the MoD's requirements process. Within the URD is expressed a Single Statement of Need. This describes, ideally in one sentence, the overarching capability gap that procurement aims address; it sets the scope for the remainder of the document. This statement is very similar to the functional purpose level of the AH in CWA; this describes the general purpose of the system: its raison d'être. Both functional purpose (CWA) and Single Statement of Need (URD) offer a short description of the system's purpose, the only difference being that the Single Statement of Need is always couched in terms of the needs of a future system, whereas CWA can be used to describe both envisioned systems and systems already in place. Furthermore, both descriptions of purpose are intentionally distinct from the technology used to achieve or satisfy them. The Single Statement of Need is described in terms of the capability need and is always described in a solution-independent manner; solution independency is a fundamental principle of the URD as a whole. The functional purpose of the system as described by the AH must also be independent of all technology; this is again a core tenet of CWA (Rasmussen *et al.* 1994, Vicente 1999, Jenkins *et al.* 2009).

Second to the Single Statement of Need are the Key User Requirements (KURs). The KURs define requirements that are central to the satisfaction of the operational need that describe essential characteristics of the required system and that set a benchmark against which to measure the performance of the system under acquisition. In these respects, they bear considerable similarity to the information in the second level down in the AH, namely the values and priority measures. The values and priority measures (or 'abstract functions' as Vicente (1999) terms them) provide the 'criteria that the work system uses for measuring its progress towards the functional purposes' (Jenkins *et al.* 2009, p. 20) and represent fundamental constraints that shape any purpose fulfilling functions or activities. Only by satisfying these criteria can a system achieve its purpose.

There are, however, some differences between the KURs and the values and priority measures. First, it is important to point out that the KURs do not summarise the entire capability need, rather they identify the primary characteristics (not all of the characteristics) that a solution must deliver. This is in contrast to the values and priority measures, which encompass the system in its entirety. Second, rather than being developed as a stand alone set of requirements, they are hand-picked from the full list of user requirements on the basis of their importance and measurability. This said, however, they do still represent a foundation for the formal validation and measurement of the system; the values and priority measures share this objective.

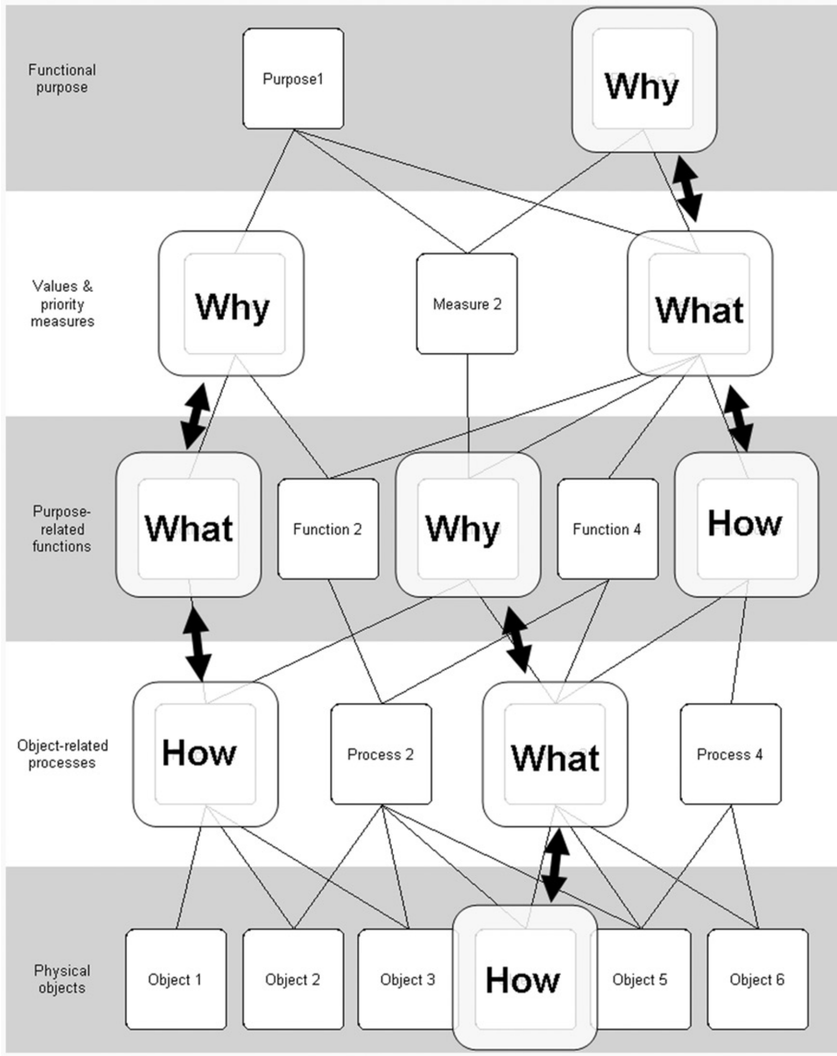

Figure 1. AH structure with means–ends causal chains overlaid.

3. WDA for requirements

3.1. *Structuring and communicating information*

CWA and requirements engineering frameworks both stipulate the need to gather information about the domain; however, it is in the structural organisation of that information that CWA can provide significant support to the requirements process. This is particularly true when considering large, complex, socio-technical systems, i.e. those systems in which the interaction between technology and human governs success (Walker *et al*. 2008, 2009). As Yu (1997) notes, as the complexity of a domain increases, so does the need for a tool to assist in the representation of that complexity. It is proposed that CWA provides the necessary set of tools to cope with this complexity (Jenkins *et al*. 2009).

The benefit provided by CWA in structuring information first comes in the earliest stage of the analysis; WDA. It is in this phase that the AH is developed. The AH provides a representation of the system that displays the underlying organisation of the work domain and of the system. Not only can this representation facilitate the requirements specification process (described in detail below), but it has the additional benefit of being easily interpreted, hence offers a means for communicating this information in a format more likely to be understood, and to be understood more quickly, than if the system description were presented in a textual format (Walker *et al.* 2010). This point refers to the adage that 'a picture is worth a thousand words'; it would likely require many pages of text to describe in equal detail the information that the AH provides about the structure of a system, without any guarantee that the interpretation would be appropriate or indeed correct. This has benefits not only for the Human Factors researcher (in developing system understanding) and the requirements engineer (in saving time and effort trying to convey the information in text), but for the individual tasked with designing the envisioned system. Though success in writing requirements is critical if the envisioned system is to provide all required capabilities, it is only one half of the process; the requirements must also be interpreted. The reliability and validity of these interpretations will not only rely on how the requirements are presented, but on their consistency also; if different levels of specificity and differing presentation methods are used within or across requirements specifications, it is unlikely that consistently reliable interpretations will follow. Though there are many sources describing the requirements analysis and specification process (e.g. Withall (2007) for software requirements or AOF (2010a, b), for the UK MoD requirements) they vary considerably in their recommendations. Furthermore, the uptake of the recommendations is not consistent across, and even within organisations. CWA, on the other hand, although requiring an element of skill from the analyst, has a well-defined structure that is generally agreed upon and adhered to within the literature. Though there have been additions and modifications suggested by various researchers (e.g. Ahlstrom 2005, Kilgore and St-Cyr 2006, Naikar *et al.* 2006), the basic process and the representations (particularly for WDA) remain the same (Vicente 1999, Jenkins *et al.* 2011).

3.2. *Functions, connections and models*

As described above, the AH is a representation of the work domain that focusses on abstract functions and functional connections; it is independent of actors and specific activities. Indeed, it is the authors contention that it is the AH that provides the most useful and applicable tool for informing the requirements specification process as it is WDA that helps think about the reasons for a system's existence (Sanderson *et al.* 1999). Furthermore, the AH specifies the domain objectives and functions that must be available and satisfied if the system is to achieve its purpose(s) (Vicente and Rasmussen 1992). As described above, the top-most level of the AH is akin to the Single Statement of Need. Developing an AH for a domain in which there is a capability gap will provide the purpose of a required system; this can then be used to inform, if not provide, the Single Statement of Need.

In the next level down from the functional purpose, the values and priority measures of the system are described. These provide two benefits when considering requirements and system design. First, though the KURs can be used to set a benchmark against which to judge system performance (described above), they do not cover all aspects of the system.

The values and priority measures, on the other hand, encompass the system in its entirety; therefore, they offer a means for measuring the effectiveness of a system as a whole and as such may be used as a form of high-level acceptance criteria. Second, and more importantly for the actual design of the requirements, they describe the high level constraints in the system. Only by satisfying these constraints can the overall system purpose be satisfied. It is for these reasons that the values and priority measures should be considered when designing a new or updated system. Not considering the nodes at this level of the hierarchy would present two significant risks; a system may be developed that cannot be validated (there would be no way to measure how well the system is achieving its purpose), or a system may be developed that does not perform within the constraints of the work domain or environment (hence does not perform at all).

Through the combination of bottom–up and top–down considerations of system design and requirements specification, it is possible to represent, in terms of functions and linkages, the functional model of a system held by experts. Rouse *et al.* (1992) suggested that the closer the parallel between system and user models, the less cognitive effort required from the user to interpret data, solve the problems and perform tasks. Given the AH is constructed through the collaborative effort of analysts, experts and representative users, it follows that the system model, if it is to be effective and valid, must comprise the functional nodes and relationships intrinsic to the work domain as described in the hierarchy (Elm *et al.* 2003). Considering and developing the AH in this manner results in a system representation that can be used as a tool to directly influence the requirements specification process such that not only is it possible to specify the requirement for a system to match the users' functional models (a requirement present in a SRD viewed by the researchers), but also allows for a description of what these functional models comprise.

4. CWA in action

Though there is a current paucity of examples within the literature that specifically describe the use of CWA in developing requirements specifications for first-of-a-kind systems, there have been a number of applications in which CWA has been applied during or before the procurement stage of the system lifecycle. Below are presented two such examples; one describes the use of CWA in providing the requirements for a training system (Naikar and Sanderson 1999, Naikar 2006), whilst the other discusses the use of CWA in evaluating design proposals for an Airborne Early Warning and Control system (Naikar and Sanderson 2001).

4.1. *Training system requirements*

Naikar (2006) describes the use of CWA in the specification of training-system requirements for the Australian Air Force's F/A-18 fighter aircraft (Naikar and Sanderson 1999, Naikar 2006). As Naikar describes, there has been a tendency in the past for organisations to assume expense equates to quality; expensive systems must, by their very nature of being expensive, be worthwhile. This is not, however, necessarily the case (Lintern and Naikar 1998). There is still a significant requirement not only for an analysis of the training needs, but of the specification of the system requirements to meet those needs. To address this need Naikar (2006) suggests using the AH, offering an explanation of the utility of each level in terms of the specification of requirements.

Table 2. Training system requirements relating to each of the five AH levels.

Functional purposes	The training system must be capable of satisfying the training objectives of the work domain
Values and priority measures	The training system must be capable of collecting data relating to the measures of performance in the work domain
Purpose-related functions	The training system must be capable of generating scenarios for training the fundamental functions of the work domain
Object-related processes	The training system must be capable of simulating the functionality of physical devices and significant environmental conditions in the work domain
Physical objects	The training system must be capable of recreating the functionally relevant properties of physical devices and significant features of the environment in the work domain

Source: Adapted from Naikar (2006).

Although the analysis was performed to specify training-system requirements, the process is equally applicable to the specification of requirements for systems across platforms and domains. Naikar's (2006) derivations of training-system requirements from each level of the AH are presented in Table 2.

An important point to make here is that the training requirements described by Naikar (2006) are slightly different in nature to the requirements set out in the URD and SRD. This was an analysis of a domain aimed at providing the functional understanding of the requirements for training, that is to say Naikar (2006) used CWA to describe the domain in such a way as to inform for which functions and activities a training system needs to train. Though this process is different from that of constructing the requirements for a system intending to fill a capability gap, the concepts remain useful. Naikar investigated a system to inform training requirements; we propose that CWA can be used to investigate a domain or environment to inform system requirements. An analysis of an operational system results in a detailed understanding of its functions and purposes, hence allows for an understanding of the functions and purposes on which training should be focussed. An analysis of a required system, and the environment in which that required system will be situated, results in a detailed understanding of the functions that will need to be performed, hence an understanding of design needs; requirements are necessarily informed.

4.2. *Tender evaluation*

Tender evaluation, or *source selection* as it is termed in the US, is the process by which a customer evaluates proposals submitted by a number of solution providers intending to build a required system. These proposals are generally constructed based on the pre-supplied requirements specification documents. In Australia, as of 2001, the primary means for judging these proposals was through technical and operational evaluation techniques (Naikar and Sanderson 2001). These techniques, Naikar and Sanderson argue, are too technically focussed. Though the methods adequately distinguish between design proposals in terms of what the proposed system components will provide, and how they are to provide it, they miss the important question of *why* those components are necessary and how individual components will interoperate within the system as a whole in terms of the overall purposes of the system. Naikar and Sanderson argued, therefore, that a framework was needed that could judge whether each proposed system could support a variety of work, satisfy the overall needs of the customer, and fulfil high level requirements of the system and the environment within which that system will be situated (Naikar and Sanderson 2001). Their solution was to use WDA, specifically the AH. Their analyses were based on the requirements documents supplied to the solution providers as well as a thorough investigation of the AEW&C environment.

There are a number of benefits in using the AH to evaluate design proposals, two of which merit particular attention. First, the AH can be used to evaluate system components based on whether or not they satisfy the priority measures of the system; as previously described, these are akin to the KURs. Second, the AH can be used to judge the success with which system components interact. Rather than acting as independent physical objects, each component, or physical device, is part of a single system that aims to support a common purpose, or set of purposes. Measuring the technical performance of each component in isolation is unlikely to be indicative of overall system performance, rather it is their interconnectedness that is critical to success (Naikar and Sanderson 2001).

In this example, the AH was developed from the system specification documents and later used to evaluate the solution providers' interpretations of those documents. Whilst this may be a slight departure from the arguments provided in this article, the link is clear; if the AH can be used to judge proposals, then it can be used to inform requirements specification. In this way, the AH is analogous to the marking scheme used by university lecturers to guide their evaluations of students' coursework. If the students are provided with the marking scheme when the coursework is set, they will likely have a better understanding of the task required of them. Similarly, if the solution provider is provided with a requirements document organised in terms of functional links, interconnections, and system priorities and purposes, they will likely design a system that better satisfies the customer's overall needs. This point was recognised by the deputy secretary of the Australian DoD who asserted that WDA should be introduced far earlier in the acquisition process than evaluation (Naikar and Sanderson 2001, p. 540).

5. The AH for interface design requirements

Though the arguments thus far have been in terms of overall system design, CWA has often been posited as a method for informing interface design (Vicente and Rasmussen 1992, Burns and Hajdukiewicz 2004); however, it is most commonly considered in the latter stages of the design cycle, after user and system requirements have already been specified. It is argued here, however, that the AH can be used much earlier in the system life-cycle, namely during the requirements specification stage. Although it may be premature to determine the specific layout of the interface at this stage of the process, it is possible to use the structure defined in the AH to provide some suggestions for interface organisation.

One system, recently having received a relatively high level of attention within the literature, serves to illustrate this principle; the mission planning system (MPS; Jenkins *et al*. 2008, Salmon *et al*. 2010a, McIlroy and Stanton 2011, Stanton and McIlroy 2010). The MPS is a software based system that allows military helicopter pilots to plan for single or multiple aircraft sortie missions and involves the management and insertion of various types of information (e.g. communication frequencies, encryption settings, flight times and localities, payload requirements, engine parameters, etc.) into a number of different windows.

In an investigation of the communications planning suite of MPS (McIlroy and Stanton 2011, Stanton and McIlroy 2010), it was determined that to perform specific functions (described on the second level from the bottom of the AH) the user required multiple windows to be open simultaneously. This equated to a need to switch between various windows when performing a single function. To address this issue, it was suggested that for each function on the object-related processes level of the AH, all of the connected physical objects should be available to the user simultaneously. Figure 2 displays an example for the function of planning for air-to-air communications, with all the relevant physical objects highlighted.

It is these objects that should be simultaneously available to the user when planning for air-to-air communications. The key concept is that the interface should be organised such that the user may perform individual functions described in the AH without the need to switch between windows or modes. This principle holds for any of the nodes on the object related processes level. For example, in Figure 3 (the full AH for MPS), it can be seen that for the planner to have an adequate understanding of all route relevant operating airspaces they must have available to them the map, the attributes of each airspace (e.g. operating hours and airspace boundary information), and information regarding their own route. Displaying these data separately will only serve to reduce usability thus negatively impacting system performance.

Though this analysis was carried out on an existing system, with the physical objects in place, it is contended that the principle remains equally applicable to systems that have yet to be defined to that level of specificity. To illustrate this point, it is again useful to turn to the MPS communications example. If the lowest level of the AH were to be absent, it would still be possible to stipulate the requirement for the simultaneous presentation of certain information. Providing the upper four layers of Figure 3 in a requirements document, alongside the stipulation that the system must be organised such that individual functions should be wholly supported by one window or mode would encourage the designer to base system organisation on users' tasks, not underlying technical architecture (as was the case in MPS; Jenkins *et al.* 2008).

This stipulation of the requirement for the functional organisation of information and options presented to the end-user would not only help prompt the user as to the information they should be entering at a given stage in the system (as information presentation is task-orientated), but support task engagement and minimise the cognitive load associated with switching between interfaces (Eggleston and Whitaker 2002). For higher level system organisation the concept is equally valid; whilst the object related process may be supported by information presented in one window or mode, it could be stipulated that the information required to fulfil higher level functions should be presented at a specific work station, terminal, control room or operating base (dependent on the size and nature of the system under development).

The use of the AH in specifying requirements would be most suitable when constructing the SRD as it considers the functional systems architecture in facilitating and clarifying the allocation of required functionality to individual systems or components (AOF 2010b). In this instance, the windows, modes or option screens made available to the user can be considered as separate components of the system. This does not describe *how* information is to be presented (an obligation of all requirements documents, military or otherwise); rather the focus is on *what* information is to be presented in concert. This provides the benefit of allowing for the specification of interface structure without providing direct guidance on the aesthetic properties of the interface; this is a task for the solution provider. Note, though, that the authors are of the opinion that this process

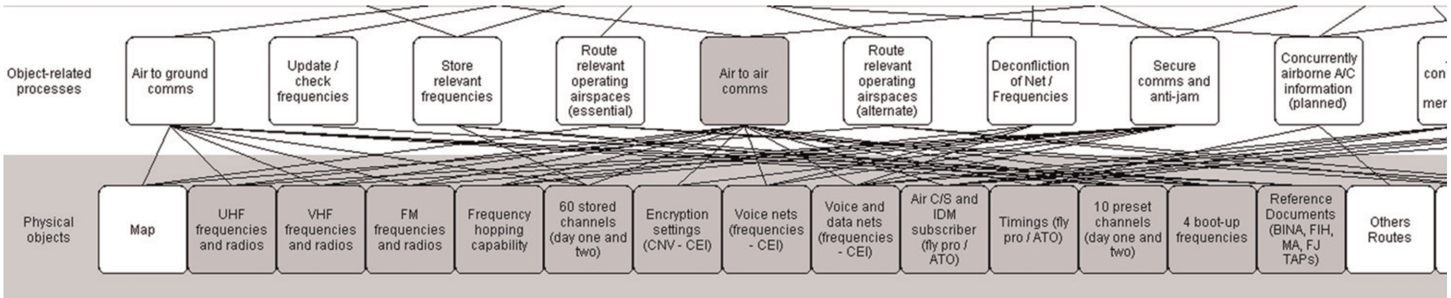

Figure 2. The function of planning for air to ground communications with all connected physical objects shaded grey.
Source: Adapted from McIlroy and Stanton (2011).

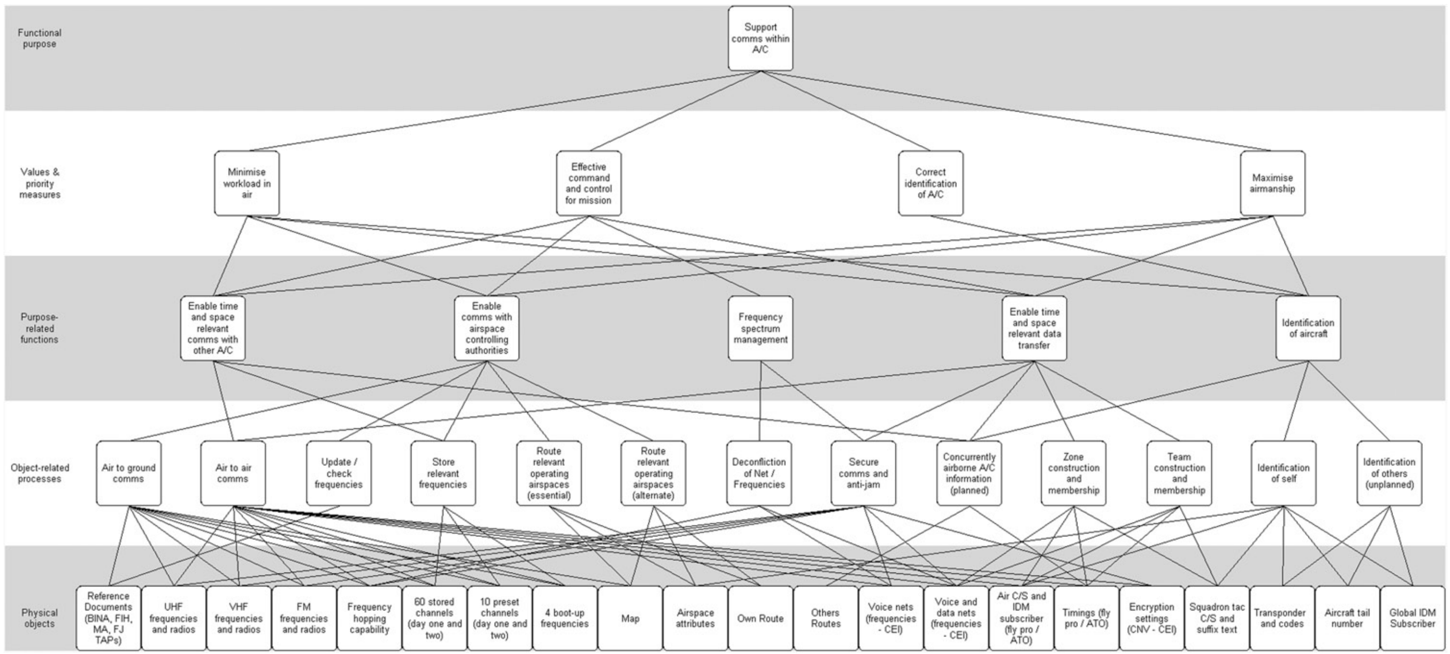

Figure 3. Full MPS abstraction hierarchy.
Source: Adapted from McIlroy and Stanton (2011).

is not ideal. Handing off requirements documents to a solution provider and waiting for a system to be tested makes the assumption that the solution provider will themselves apply Human Factors methods in design, supplying an interface that is optimally designed to support the task. This has not always been the case in systems acquired by the UK MoD (Salmon *et al.* 2010b).

6. Function-based worker competencies analysis

The focus, so far, has been on the use of the AH to inform requirements; however, this is not to say that the remaining phases are not of use. Of particular use in the specification of requirements is Worker Competencies Analysis (WCA), the final stage of CWA. WCA addresses the constraints dictating actor behaviour for different system activities (Jenkins *et al.* 2009) and is commonly modelled using Rasmussen's (1983) skills, rules and knowledge taxonomy. Previous analyses of worker competencies have commonly adopted a decision ladder approach, i.e. a skills, rules and knowledge inventory is constructed for each control task, with each row describing the different types of behaviour required for individual steps on decision ladders for that control task (Kilgore and St-Cyr 2006, Kilgore *et al.* 2009). We contend that although this process is indeed useful in the latter stages of the design process, it is not a method that can be applied in requirements specification (McIlroy and Stanton 2011). Describing competencies in this way requires the system to have had activities described and analysed. It is not possible to construct decision ladders for systems in which no specific activities (and strategies therein) have yet to be described. This does not, however, prevent this form of analysis providing an input in early stages of requirements analysis; rather the approach must be adapted. McIlroy and Stanton (2011) provide a description of the SRK inventory informed by nodes taken from the AH rather than decision ladder steps, hence a function-based approach. In their argument, they contend that using the SRK inventory in this way allows for a description of the competencies required from system actors before a particular solution has been defined.

The process of constructing the SRK inventory in this manner requires a considerable volume of expert and user-input. Once this input is supplied, however, converting the information presented in the inventory into system requirements incurs minimal time and effort. The resulting requirements can be used not only to inform the design of the interface, but inform personnel recruitment requirements and training requirements (should the skills and knowledge be unavailable in the current personnel base). As an example, consider the function 'deconfliction of nets/frequencies' in the WCA carried out in the investigation of the communications suite of MPS (Table 3; McIlroy and Stanton 2011). Deconfliction of nets and frequencies refers to the need to ensure the communication frequencies and 'nets' (specific to data communications) used for a specific sortie do not interfere with each other. Information regarding conflictions is contained within a matrix of frequencies; this matrix is available to the pilot for reference. Should conflicting nets or frequencies be selected, the security and reliability of communications on those channels will be compromised.

In terms of interface design, for the skill-based behaviour, the requirement specified may be worded as such; 'The user shall always be able to ensure used frequencies and nets do not interfere with each other'. It is necessary to explicitly state that this must always be possible as it is a requirement of all users, novice to expert. For the

Table 3. Skill-, rule- and knowledge-based behaviours required to perform the function 'deconfliction of net/frequencies'.

Purpose-related functions	Skill-based behaviour	Rule-based behaviour	Knowledge-based behaviour
Deconfliction of nets/frequencies	Ensure used frequencies/nets do not interfere with each other	From table, see which frequencies interfere with each other and avoid those combinations	Know how frequencies may interfere with each other and the consequences of that interference. Ensure this does not happen by selecting frequencies that are of a sufficient separation

Source: Adapted from McIlroy and Stanton (2011).

remaining two behaviours, however, it should be explicitly stated that information should only be available upon user request; having this information continually presented would not provide any benefit to the expert user as it is predominantly skill they use to guide actions (see Rasmussen 1983, for a discussion on the types of behaviour governing actions of expert and novice actors). Only less experienced users would require this information, therefore wording of requirements should make this explicit. For example, the requirement specified to support rule-based behaviour may be as follows: 'the system shall provide a table, upon user request, that provides information displaying the frequencies and nets that interfere with each other'. Similarly, to support knowledge-based behaviour: 'the system shall provide information, upon user request, regarding the confliction of nets and frequencies, for why these conflictions arise, and offer selection options that are of sufficient separation as to avoid the problem of confliction'. Specifying the interface in this way avoids this issue of determining *how* an interface is to be designed by simply stating *what* information is to be presented, thereby adhering to the fundamental principle of specific solution independency (once again, however, the authors would like to make their position clear that they believe this process to be sub-optimal).

This method of specifying requirements is, in effect, allowing for the description of the information an actor needs to perform a task. This type of information requirements specification can be contained within the SRS or within SRDs, though often is not, and even when it is, it is rarely specified to the same level of detail. Of the requirements documents viewed by the analysts, only one had a sufficiently extensive section on Human Factors considerations; however, though the requirements contained in the section were beneficial, they did not, in the analysts' judgement, go into enough detail when describing interface design requirements. The example presented below, again viewed by one of the analysts in a SSRD, represents the most specific of all the requirements analysed with regard to information requiring presentation:

> The information presented and the style of its display should reflect what the operator needs to perform the task.

This requirement assumes the solution provider has a sufficiently detailed understanding of the users' tasks to provide an interface that optimally supports work. As the solution provider is rarely, if ever, the end-user, there should be no reason to make this assumption. Without an understanding of the task, it will be impossible to design a system that optimally supports it. Furthermore, the ill-defined, somewhat vague nature of this requirement is inherently disadvantageous in two primary ways: (1) it does not constrain the solution provider in a concrete way and so may have no effect on interface design and (2) without the addition of an explicit description of the task (which is outside the scope of the requirements documents), it is immeasurable and unfalsifiable, therefore providing no benefit and offering no means for judging the efficacy or success of the acquired system.

7. The importance of process

Although CWA provides pictorial and graphical representations of the system that can, on their own, be used to inform requirements specification, it is the process of performing the analysis that provides the greatest benefit. In order to have a solid understanding of the system under consideration, it is crucial that the customer (i.e. the organisation procuring the system) is involved throughout the process. Involvement must include not only the customer however, but Human Factors specialists, domain experts and representative users of the envisioned system. This collaborative process between the customer and industry is not strictly participatory design (e.g. Schuler and Namioka 1993), but a participatory analysis of need, that is to say the needs of the users and the system. Furthermore, it is critical that each party is involved at every stage; it is not sufficient to create a CWA output in one session and expect that to be satisfactory and lasting. The process is inherently iterative, requiring a number of rounds of development and evaluation of analyses. Moreover, any changes made to any of the outputs will likely have an effect in other parts of the analysis, therefore requiring validation by all involved. This point is succinctly described in Elm *et al.* (2003):

> The artefacts serve as a post hoc mechanism to record the results of the design thinking and as stepping stones for the subsequent step of the process. Each intermediate artefact also provides an opportunity to evaluate the completeness and quality of the analysis/design effort, enabling modifications to be made early in the process. The linkage between artefacts also ensures an integrative process; changes in one artefact cascades along the design thread necessitating changes to all. (p. 7)

Admittedly, it is not always practical to conduct a full CWA on an envisioned system in the earliest stages of its life-cycle. If a system is still in the concept stage the activities to be performed may not yet have been described, hence the Control Task Analysis, the Strategies Analysis and the Social Organisation and Cooperation Analysis stages may be quite effortful to construct, and as such incur more time and cost than the customer may be willing to offer. It is argued, then, that in these instances a CWA should be conducted on the system that is currently in place, that is to say the system that will be replaced, or on systems performing similar purposes. The descriptions of activity, strategies and social organisation will still be applicable to the proposed or envisioned system, therefore providing a basis for development of the new system, unless the system under acquisition is intending to provide a capability that is currently unavailable, involving completely new roles and activities. As this is rarely the case, the full CWA can be applied in most instances; in the few instances where this is the case, it is proposed that Work Domain and Worker Competencies Analyses will still be applicable and beneficial in relation to requirements specification.

It has been previously noted that, in the authors' judgement, the process of reducing customer involvement after requirements documents have been passed off to industry cannot ensure a system that is optimally designed for use. It is argued that the full CWA process should be used as a support tool throughout the system's life-cycle; with the AH and WCA providing considerable benefit in the requirements stages, and the remaining three phases applicable throughout design and manufacture. It is argued that CWA, applied correctly by individuals of sufficient expertise in Human Factors, can be used to structure and organise the collaborative and iterative process of design and manufacture as long as the customer and a representative sample of domain experts and prospective end-users remain involved. Furthermore, as described in Section 3.1, the AH provides a graphical representation that summarises the system in an easily interpreted format. This can also be said for the other CWA outputs; each provides a different representation of the system that, if replicated in text, would require an extensive written account to describe. As with the AH, not only can the representations offered by CWA facilitate system understanding, they can provide a means for communicating that information to those responsible for designing, and ultimately building (hardware) or coding for (software) that system.

8. Discussion

The primary conclusions of the exploration of CWA as a tool for informing requirements specification are that not only does CWA bear significant resemblance to some requirements engineering frameworks, but that CWA can offer a number of ways of informing, if not supplying user and system requirements. Although the CWA process comprises five stages, it is argued that the AH constructed in the WDA stage and the AH informed SRK inventory developed in the Worker Competencies stage are the most applicable. Despite the minimal discussion of the other stages of CWA, namely Control Task Analysis, Strategies Analysis and Social Organisation and Cooperation Analysis, it is argued that they are equally important when designing new systems. It is merely contended that they are not as applicable in the requirements specification stage of the system life-cycle; rather they are of more benefit during the actual design process. Though the requirements documents aim to remain solution independent, it is the authors' contention that the customer should retain direct involvement throughout the design process, applying Human Factors methods throughout (Stanton *et al*. 2005a, b). This is not the current standard practice, certainly within the UK MoD; it is not uncommon that once requirements documents are constructed and passed down to industry, the customer ends its involvement until some pre-defined testing of prototype systems. At this stage, it is often too late to provide useful design guidance; it is much more efficient and effective to change design before a system is hard built (Stanton and Young 1999).

An important point to make relates to the importance of the customer developing and maintaining a solid understanding of the capability and the domain and environment in which that capability gap resides. Without fully understanding the need of the envisioned system, it is impossible to specify requirements down to a sufficient level of detail such that the delivered system will perform to an acceptable standard. The lack of domain understanding often held by the customer is well-illustrated by this example from a UK MoD's SSRD viewed by the analysts. The two requirements presented below were presented in the SSRD one after the other, in the order they are offered here (note that though it is industry that writes these documents, they are in fact owned by the MoD, and hence require MoD approval). They provide a typical example of the MoD's allowance of industry to set their own goal posts:

> The user/task performance shall be shown to be achieved within specified criteria (such as timings and acceptable error rates). The contractor is to develop the user/task performance criteria.

Though it would be beneficial for organisations procuring externally designed systems to be able to trust the solution provider to self-regulate (be it the defence industry, software companies, communication companies, etc.), as it is profit that drives business, this trust could be misplaced. As Gray (2009) points out in relation to the costing of projects in the UK:

> Simply put, many participants in the procurement system have a vested interest in optimistically mis-estimating the outcome. (Gray 2009, p. 19)

Moreover, it is especially important for the system procurer to retain control over its acquisition process given the controversy surrounding overspend, under-performance of delivered systems and delays in delivery, both in the UK and US military (Isenberg 2007, Gray 2009) and across public sectors (see Hulme 1997, for a discussion of procurement failure in Management Information Systems projects in the US; see Walker *et al.* 2009, p. 176, for a summary of failures in system procurement across organisations in the UK). Control can only be retained, and suitably asserted, if the customer has a solid understanding of the capability need, the domain in which the need resides, and the functional models of the prospective end-users; CWA, if successfully applied, can provide this information.

9. Conclusions

In summary, this discussion of CWA and requirements has three major conclusions in terms of requirements specification, and has three major conclusions in terms of the issues with the current requirements and acquisition process. In terms of the specification of requirements, the three key points are as follows: (1) CWA's AH can be used to help understand the domain of interest in terms of its constraints, the opportunities it affords and the functional models held by domain experts and prospective end-users; (2) the communication of information through a graphical or pictorial format provides significant benefit, in terms of ease of understanding, over the use of textual descriptions of systems and (3) the AH and Worker Competencies Analyses can inform requirements specification, and, in some cases, can directly provide requirements that need only minimal adaptation. In terms of the issues with the current process, the three central conclusions are as follows, primarily with respect to the process adopted by the UK MoD (it was to this process the authors had access) though applicable to any organisation that out-sources for system procurement: (1) the customer does not always determine its own success criteria, rather they leave it to industry to regulate themselves; (2) the iterative process of requirements analysis, specification and design is not always adhered to, with Human Factors analyses only applied during the earliest stages, if at all. Although it is critical that Human Factors play a role at this early stage, it is of equal importance that the process is carried throughout the design stages; the process is inherently iterative and (3) the customer does not always retain involvement throughout the requirements or design phases, rather it is common for the procuring organisation to take a backseat once requirements documents have been handed to industry and only become involved once more when a testable prototype has been produced. Indeed, it is the authors experience in talking to those involved in the process that the UK MoD does not always have even this level of involvement, with many of the requirements documents being written by the companies paid to produce the solution. Though this approach may be the most cost effective in the short-term, it is clear that this system of requirements production and system procurement is not performing at an acceptable level of expense, in terms of both time and money (Humphries and Wilding 2001, Gray 2009). Though it is true that industry often possess a higher degree of expertise relating to the domains in which systems will be deployed, it is important that customers do not pass all responsibility to them; as previously mentioned, it may not be in the best interest of the customer to allow the system provider to self-regulate. This is particularly important in the current economic climate; only with full control over their acquisition process will government bodies have the ability to reduce the budget deficits forecast to mount up over next 10 years (National Audit Office 2009). Finally, it is important to note that whilst this article presents a discussion of *how* CWA could, and in the authors' judgement should be used in requirements specification, it does not present a full example of a system specification based on CWA; this presents a critical avenue for future work. An example of such an application is clearly needed if government and industry are to be persuaded that CWA can truly provide all the benefits proposed both in this article and by CWA practitioners more widely.

Acknowledgements

This research from the Human Factors Integration Defence Technology Centre was part-funded by the Human Sciences Domain of the UK Ministry of Defence Scientific Research Programme. Any views expressed are those of the author(s) and do not necessarily represent those of the MoD or any other UK government department.

References

Ahlstrom, U., 2005. Work domain analysis for air traffic control weather displays. *Journal of Safety Research*, 36, 159–169.

AOF, 2010a. Acquisition Operating Framework, *Requirements and acceptance* [online]. Available from: http://www.aof.mod.uk/aofcontent/tactical/randa/index.htm [Accessed 24 May 2010].

AOF, 2010b. Acquisition Operating Framework, *Defining system requirements* [online]. Available from: http://www.aof.mod.uk/aofcontent/tactical/randa/content/srdosanddonts.htm [Accessed 25 May 2010].

Borgida, A., Greenspan, S., and Mylopoulos, J., 1985. Knowledge representation as the basis for requirements specifications. *Computer*, 18, 82–91.

Burns, C.M. and Hajdukiewicz, J.R., 2004. *Ecological interface design*. Boca Raton, FL: CRC Press.

Cheng, B. and Atlee, J., 2007. Research directions in requirements engineering. *In*: L. Briand and A. Wolf, eds. *Future of software engineering 2007*. Los Alamitos, CA: IEEE-CS Press, 285–303.

Curtis, B., Krasner, H., and Iscoe, N., 1988. A field study of the software design process for large systems. *Communications of the ACM*, 31, 1268–1287.

Dardenne, A., Fickas, S., and van Lamsweerde, A., 1991. Goal-directed concept acquisition in requirements elicitation. *In*: *Proceedings of the IWSSD-6-6th international workshop on software specification and design*, Como, Italy, 14–21.

Eggleston, R.G. and Whitaker, R.D., 2002. Work-centred support systems design: using organizing frames to reduce work complexity. *In*: *Proceedings of the human factors and* *ergonomics society 46th annual meeting*. Santa Monica, CA: Human Factors and Ergonomics Society, 265–269.

Elm, W.C., *et al.*, 2003. Applied cognitive work analysis: a pragmatic methodology for designing revolutionary cognitive affordances. *In*: E. Hollnagel, ed. *Handbook for cognitive task design*. London: Lawrence Erlbaum Associates, 653–678.

Ernst, N.A., Jamieson, G.A., and Mylopoulos, J., 2006. Integrating requirements engineering and cognitive work analysis: a case study. *In*: *Presented at the fourth conference on systems engineering research*, Los Angeles, CA.

Gray, B., 2009. *Review of acquisition for the secretary of state: an independent report by Bernard Gray*. London: The Stationary Office.

Hollnagel, E. and Woods, D.D., 1999. Cognitive systems engineering: new wine in new bottles. *International Journal of Human-Computer Studies*, 51, 339–356.

Hsia, P., Davis, A., and Kung, D., 1993. Status report: requirements engineering. *IEEE Software*, 10, 75–79.

Hulme, M.R., 1997. Procurement reform and MIS project success. *International Journal of Purchasing and Materials Management*, 33, 2–15.

Humphries, A.S. and Wilding, R., 2001. Partnerships in UK defence procurement. *The International Journal of Logistics Management*, 12, 83–96.

Isenberg, D., 2007. *Budgeting for the empire: the effect of Afghanistan on military forces, budgets, and plans*. Oakland, CA: The Independent Institute.

Jackson, M. and Zave, P., 1993. Domain descriptions. *In*: *International symposium on requirements engineering*, San Diego, CA, 56–64.

Jenkins, D.P., *et al.*, 2008. Using cognitive work analysis to explore activity allocation with military domains. *Ergonomics*, 51, 798–815.

Jenkins, D.P., *et al.*, 2009. *Cognitive work analysis: coping with complexity*. Farnham, England: Ashgate.

Jenkins, D.P., *et al.*, 2011. Using work domain analysis to evaluate the impact of technological change on the performance of complex socio-technical systems. *Theoretical Issues in Ergonomics Science*, 12 (1), 1–14.

Kilgore, R. and St-Cyr, O., 2006. SRK inventory: a tool for structuring and capturing worker competencies analysis. *In*: *Proceedings of the human factors and ergonomics society 50th annual meeting 2006*, San Francisco, CA, 506–509.

Kilgore, R., St-Cyr, O., and Jamieson, G.A., 2009. From work domains to worker competencies: a five-phase CWA. *In*: A.M. Bisantz and C.M. Burns, eds. *Applications of cognitive work analysis*. Boca Raton, FL: CRC Press, 15–47.

Leveson, N.G., 2000. Intent specifications: an approach to building human-centred specifications. *IEEE Transactions on Software Engineering*, 26, 15–36.

Lintern, G. and Naikar, N., 1998. Cognitive work analysis for training system design. *In*: *Proceedings of the Australasian computer human interaction conference (OzCHI'98)*. Los Alamitos, CA: IEEE Computer Society Press, 252–259.

McIlroy, R.C. and Stanton, N.A., 2011. Getting past first base: going all the way with cognitive work analysis. *Applied Ergonomics*, 42, 358–370.

Naikar, N., 2006. Beyond interface design: further applications of cognitive work analysis. *International Journal of Industrial Ergonomics*, 36, 423–438.

Naikar, N., Moylan, A., and Pearce, B., 2006. Analysing activity in complex systems with cognitive work analysis: concepts, guidelines, and case study for control task analysis. *Theoretical Issues in Ergonomics Science*, 7, 371–394.

Naikar, N. and Sanderson, P.M., 1999. Work domain analysis for training system definition and acquisition. *International Journal of Aviation Psychology*, 9, 271–290.

Naikar, N. and Sanderson, P., 2001. Evaluating design proposals for complex systems with work domain analysis. *Human Factors*, 43, 529–542.

National Audit Office, 2009. *Ministry of defence: the major projects report 2009*. London: The Stationary Office.

Nuseibeh, B. and Easterbrook, S., 2000. Requirements engineering: a roadmap. *In*: *The future of software engineering, special issue 22nd international conference on software engineering*, ACMIEEE, 37–46.

Rasmussen, J., 1983. Skills, rules, knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 234–243.

Rasmussen, J., Pejtersen, A., and Goodstein, L.P., 1994. *Cognitive systems engineering*. New York: Wiley.

Rouse, W.B., Cannon-Bowers, J.A., and Salas, E., 1992. The role of mental models in team performance in complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 1296–1308.

Salmon, P.M., *et al*., 2010a. Hierarchical task analysis vs. cognitive work analysis: comparison of theory, methodology and contribution to system design. *Theoretical Issues in Ergonomics Science*, 11, 504–531.

Salmon, P.M., *et al*., 2010b. Decisions, decisions . . . and even more decisions: evaluation of a digitized mission support system in the land warfare domain. *International Journal of Human-Computer Interaction*, 26, 206–227.

Sanderson, P., *et al*., 1999. Use of cognitive work analysis across the system life cycle: from requirements to decommission. *In*: *Proceedings of the 43rd annual meeting of the Human Factors and Ergonomics Society*. Santa Monica, CA: Human Factors and Ergonomics Society, 318–322.

Schraagen, J.M., Chipman, S.F., and Shalin, V.J., 2000. *Cognitive task analysis*. *Expertise: research and application series*. Mahwah, NJ: Lawrence Erlbaum Associates.

Schuler, D. and Namioka, A., 1993. *Participatory design: principles and practices*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stanton, N.A., 2006. Hierarchical task analysis: developments, applications and extensions. *Applied Ergonomics*, 37, 55–79.

Stanton, N.A. and McIlroy, R.C., 2010. Designing mission communication planning: the role of rich pictures and cognitive work analysis. *Theoretical Issues in Ergonomics Science*, iFirst, DOI: 10.1080/1463922X.2010.497197.

Stanton, N.A. and Young, M.S., 1999. What price ergonomics? *Nature*, 399, 197–198.

Stanton, N.A., *et al*., 2005a. *Handbook of human factors and ergonomics methods*. London: CRC Press.

Stanton, N.A., *et al*., 2005b. *Human factors methods: a practical guide for engineering and design*. Aldershot: Ashgate.

Stanton, N.A., *et al*., 2009. *Digitising command and control: a human factors analysis of mission planning and battlespace management*. Aldershot: Ashgate.

van Vliet, H., 2000. *Software engineering: principles and practice*. New York: John Wiley & Sons.

Vicente, K.J., 1999. *Cognitive work analysis: toward safe, productive, and healthy computer-based work*. Mahwah, NJ: Lawrence Erlbaum Associates.

Vicente, K.J. and Rasmussen, J., 1992. Ecological interface design: theoretical foundations. *IEEE Transactions of Systems, Man and Cybernetics*, 22, 589–606.

Walker, G.H., *et al*., 2008. A review of sociotechnical systems theory: a classic concept for new command and control paradigms. *Theoretical Issues in Ergonomics Science*, 9, 479–499.

Walker, G.H., *et al*., 2009. *Command and control: the sociotechnical perspective*. Aldershot: Ashgate.

Walker, G.H., *et al*., 2010. From ethnography to the EAST method: a tractable approach to representing distributed cognition in Air Traffic Control. *Ergonomics*, 53, 184–197.

Withall, S., 2007. *Software requirement patterns*. Redmond: Microsoft Press.

Yu, E.S., 1997. Towards modelling and reasoning support for early-phase requirements engineering. *In*: *International symposium on requirements engineering*, 6–10 January, Annapolis, MD, 226–235.

About the authors

Rich C. McIlroy is a Human Factors researcher at the University of Southampton's Transportation Research Group. He obtained his BSc (Hons) in Psychology in 2008 and his MSc in Research Methods in Psychology in 2009, both from the University of Southampton. His current research interests include the use of Cognitive Work Analysis in the investigation of complex socio-technical systems and the use of verbal reports both as a method of information acquisition and as a means for investigating expertise.

Neville A. Stanton has a BSc (Hons) in Occupational Psychology from the University of Hull, an MPhil in Applied Psychology and a PhD in Human Factors from the Aston University in Birmingham. He holds a Chair in Human Factors in the Transportation Research Group within the School of Civil Engineering and the Environment at the University of Southampton. Professor Stanton is a Fellow and Chartered Psychologist registered with the British Psychological Society and the Health Professions Council, a Fellow of The Ergonomics Society and a member of the Institution of Engineering and Technology. He has published over 150 peer-reviewed journal papers and 19 books on Human Factors and Ergonomics. In 1998, he was awarded the Institution of Electrical Engineers Divisional Premium Award for a co-authored paper on Engineering Psychology and System Safety. The Ergonomics Society awarded him the Otto Edholm medal in 2001 and The President's Medal in 2008 for his contribution to basic and applied ergonomics research. In 2007, The Royal Aeronautical Society awarded him and his colleagues the Hodgson Medal and Bronze Award for their work on flight deck safety. Professor Stanton is an editor of the journal *Ergonomics* and on the editorial board of *Theoretical Issues in Ergonomics Science*.