Ergonomics
Vol. 49, No. 7, 10 June 2006, 688–705

Taylor & Francis
Taylor & Francis Group

Evolvable work-centred support systems for command and control: creating systems users can adapt to meet changing demands

E. ROTH‡*, R. SCOTT†, S. DEUTSCH†, S. KUPER§, V. SCHMIDT§, M. STILSON§, and J. WAMPLER§

†BBN Technologies, Cambridge, MA, USA
‡Roth Cognitive Engineering, Brookline, MA, USA
§Air Force Research Laboratory, Wright–Patterson AFB, OH, USA

Military command and control (C2) organizations are complex socio-technical systems which must constantly adapt to meet changing operational requirements. We describe our experiences in developing a work-centred support system (WCSS) to aid weather forecasting and monitoring in a military airlift C2 organization as an illustrative case. As part of the development process we conducted field observations both before and after introduction of the WCSS in their operations centre. A striking finding was the constant changes that operations personnel faced (changes in goals and priorities, changes in scale of operations, changes in team roles and structure, and changes in information sources and systems). We describe the changes in workplace demands that we observed and the modifications we needed to make to the WCSS in response. For today's fielded systems, it is seldom possible to make changes that are responsive to users' changing requirements in a timely manner. We argue for the need to incorporate facilities that enable users to adapt their systems to the changing requirements of work and point to some promising directions towards evolvable work-centred support systems.

Keywords: Work-centred support systems; Evolvable systems; End-user development; C2; Weather forecasting; Work-centred design

1. Introduction

Military command and control (C2) organizations are complex socio-technical systems which must constantly adapt to meet changing operational requirements. Geopolitical

*Corresponding author. Email: emroth@mindspring.com

Ergonomics
ISSN 0014-0139 print/ISSN 1366-5847 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00140130600612556

changes, organizational changes, and opportunities to exploit new information sources drive the need for rapid changes in software support systems to keep pace with the changing character of work. Unfortunately, military C2 organizations often operate in an environment supported by inflexible software systems. Even simple user change requests can take months to be satisfied. The life-cycle of a change request, from prioritization and assignment, through development, test, evaluation, and certification, to deployment can significantly lag behind the pace at which work demands shift (Truex *et al*. 1999). In this paper we argue for the need to develop new design philosophies and tools to better support the evolving nature of work, and to suggest an approach that fulfils this requirement.

Over the past several years we have been developing work-centred support systems (WCSS) to aid mission planning and C2 in a military airlift service organization (Scott *et al*. 2002, 2005, Wampler *et al*. 2005). WCSS are designed to provide comprehensive support for the multiple aspects of work (e.g. decision support, product development support, collaborative support, and work management support) within an integrated work-oriented framework (Eggleston *et al*. 2000, Eggleston and Whitaker 2002, Eggleston 2003).

Our first system, the Work-Centred Support System for Global Weather Management (WCSS-GWM), was developed to support weather forecasting and monitoring and is currently installed and in use in the operations centre of the airlift service organization (Scott *et al*. 2005). More recently, we have been working on expanding the scope of support to cover C2 of mission flights more broadly, focusing on the processes involved in monitoring for and responding to unexpected changes that arise during execution of mission flights (Wampler *et al*. 2005).

As part of the work-centred design process, we had the opportunity to perform field observations and structured interviews with weather forecasting and C2 personnel over a span of 4 years. Field observations were conducted in the operations centre both before development of our initial design concepts, so as to ground the design in the field of practice, as well as after the initial system was deployed (towards the end of the second year), so as to ensure that elements of work that were unanticipated and not well supported would be uncovered and addressed (Woods 1998, Woods and Dekker 2000).

Over the course of 4 years of research we observed a wide range of changes that impacted cognitive and collaborative work in the C2 operations centre. We describe the changes that we observed, the informal artefacts that users created, and the requests for modifications to the WCSS-GWM that users made in response to these changing demands. Our findings are presented as a case study to illustrate the challenges confronted in designing systems to support a constantly changing C2 environment. The results point to the need for software systems which can evolve to adapt to the inevitable changes that arise in the world. We coin the term 'evolvable work-centred support systems' to describe the kinds of adaptable work-centred systems we envisage, and point to some promising software directions for achieving that aim.

A distinguishing feature of an evolvable work-centred support system is that its very derivation is based on a work-centred perspective. We applied the same work-centred design methodology, grounded in an analysis of the demands of work, to derive the requirements for evolvable work-centred support systems.

We begin by introducing WCSS concepts and describing the WCSS-GWM that we developed for the C2 organization that embodied this design philosophy (section 2). We then present the results of two analyses that we conducted to examine the operational changes that occurred in the airlift organization C2 operations centre over the 4 year span

of the study and their impacts (section 3). The first analysis took a broad look at the kinds of work-arounds and informal artefacts developed by different groups within the C2 operations centre to compensate for the inability of existing software systems to accommodate operational changes. The second analysis focused more narrowly on users of the WCSS-GWM. We examined change requests that were submitted to the WCSS-GWM software design team by the user community to understand what motivated the change request and what changes needed to be made to the WCSS-GWM to accommodate the request. The results of these two analyses reveal the wide range of changes that was experienced by the organization and their consequences. The results argue for a need for evolvable systems, and point to the kinds of capabilities that evolvable work-centred support systems need to include.

Section 4 explores software technologies that can provide the underpinnings for development of evolvable work-centred systems. In the final section we discuss evolvable work-centred systems in the context of other similar calls for systems that can more readily adapt to unanticipated change.

2. Developing a work-centred support system for command and control

The WCSS-GWM was developed to support weather forecasting and monitoring in a military airlift service organization. It employed the work-centred design (WCD) methodology and was intended to provide an illustration of a WCSS in a military C2 organization (Eggleston *et al*. 2000, 2003, 2005, Eggleston and Whitaker 2002, Scott *et al*. 2002, 2005, Eggleston 2003). WCD is consistent with socio-technical principles in its emphasis on the interdependency of technology and work organization and the importance of analysing situated work in the context of the broader social organization (Cherns 1987). A holistic approach is taken to the analysis and support of work which includes consideration of the following.

1. **Decision support**: aiding problem solving and other cognitive processes in the process of performing work.
2. **Product development support**: aiding production of the deliverable artefact(s) of work.
3. **Collaborative support**: aiding team and colleague interactions in work.
4. **Work management support**: aiding the metacognitive activities entailed in prioritizing and managing the multiple interwoven tasks that normally arise in work.

A fundamental aspect of WCD is analysis and modelling of the **work ecology** to uncover the elements of work that require support. The process starts with knowledge capture methods such as ethnographic field observations and structured interview techniques (Militello and Hutton 1998, Roth and Patterson 2005) to uncover the goals and constraints in the work domain, the work requirements, the sources of complexity, and the cognitive and collaborative demands entailed. Formal methods can then be employed to represent the results of the analysis. These include work domain analysis methods which model the intrinsic characteristics of the work to be achieved (Vicente 1999, Elm *et al*. 2003), as well as methods that model workflow dynamics within and across individuals and groups required to achieve work goals (Kirwan and Ainsworth 1992) and use scenarios that illustrate particular work threads requiring support (Carroll and Rosson 1992).

The products of work ecology analysis define the work-aiding requirements to support domain practitioners in performing work in a flexible and adaptable manner, given the

dynamics of the work context. These requirements are used to guide work-aid design which involves development and prototyping of work-aiding concepts. Work aiding may take the form of **representational aiding** (Woods and Roth 1988), which is provided through the use of work domain visualizations or **direct aiding** provided by a coordinated set of software agents that interact with the user and are clearly connected to or are embedded in the work domain visualizations (Eggleston 2003, Scott *et al.* 2005).

The WCSS-GWM was developed to support weather forecasting and monitoring in a military airlift service organization. It employed the work-centred design methodology and was intended to provide an illustration of a work-centred support system in a military C2 organization.

The WCSS-GWM was developed as part of an ongoing R&D programme to develop WCSS for the military airlift service organization. The airlift service organization serves as a C2 hub which plans, schedules, and tracks airlift missions to move military equipment and personnel worldwide. It is a global command centre with several hundred people planning and executing approximately 350 missions per day. It includes an operations centre that is responsible for the monitoring and control of flight missions starting 24 hours before planned mission launch through to completion of the mission. The operations centre is manned by duty officers (DOs), enlisted controllers, and flight managers (FMs) who are responsible for identifying, tracking, and resolving problems. They are supported by a number of additional organizational groups providing specialized expertise including a group responsible for weather forecasting and tracking.

Traditionally, airlift pilots have been responsible for their own flight planning, including obtaining pre-flight weather briefings. In this organization, a new approach was initiated to reduce the amount of time an aircrew had to devote to these tasks. A FM position was created with the primary responsibility for planning and managing multiple flights, both pre-flight and en route. This includes obtaining a weather briefing and providing the pilot with a complete flight plan, including weather forecast information. The FM is viewed as a 'virtual crew member' in support of the pilot. Weather can significantly influence pre-flight and en route flight management decisions (e.g. there may be a need to accelerate, delay, or re-route a flight because of unfavourable weather conditions). As a result, weather forecasters must work closely with the FMs to evaluate weather conditions at the departure and arrival airfields as well as along the planned route. The focus of our effort was on developing a WCSS to aid near-term weather forecasting in support of planning and managing airlifts, both pre-flight and en route.

At the time that the study was initiated (February 2001), FMs and weather forecasters worked closely together to determine the potential impact of predicted weather on the viability of upcoming flights. If hazardous weather conditions (e.g. high turbulence or lightning) were forecast, the FM and weather forecaster worked collaboratively to identify alternative routing that would avoid the problematic weather areas. However, they had limited software tools to support their collaborative decision-making processes. While the weather forecasters had various displays available for actual and predicted weather in different parts of the world, the information came from many sources and was presented on separate displays. Further, there were no graphics depicting the planned flight paths of upcoming missions, making it necessary for forecasters and FMs to mentally fuse the various sources of disparate information to assess the potential impact of weather on a mission.

The WCSS-GWM was designed to address this problem. It provides integrated visualizations that superimpose mission flight plans with real-time and forecast weather

information on a geospatial display so as to enable weather forecasters and FMs to directly 'see' the impact of weather on flight missions. It also includes intelligent software agents which monitor weather conditions and missions. Figure 1 shows a screenshot of the WCSS-GWM display depicting the ability to create and modify software agents. A full description of the WCSS-GWM is provided elsewhere (Scott *et al*. 2005).

The WCSS-GWM was developed using the WCD methodology. The WCSS-GWM development programme covered 3 years from gathering initial requirements to delivery of a 24×7 operational system. The first year was largely devoted to gaining an initial understanding of the domain, the systems supporting the present-day workflow, and exploration of the possible work-centred support systems that might be implemented. Field observations in the operations centre and interviews with flight managers, weather forecasters, and other operations centre personnel were used to define initial aiding concepts. Initial concepts were refined based on user feedback of rapid prototypes as well as a formal user evaluation study which was completed in the second year of development (Eggleston *et al*. 2003). A functioning system was installed in the operations centre subsequent to the evaluation. It became part of the standard set of displays used by weather forecasters to make forecasts and monitor weather for potential impact on missions. Feedback from users guided the refinement of the system over the third year, as the system was completed.

A fourth year has elapsed during which time we have conducted additional observations and interviews in the C2 operations centre. These field observations were performed both as a means of obtaining additional feedback on the integration of the WCSS-GWM into the flight planning and management process, and as part of activities to define additional WCSS for the operations centre.

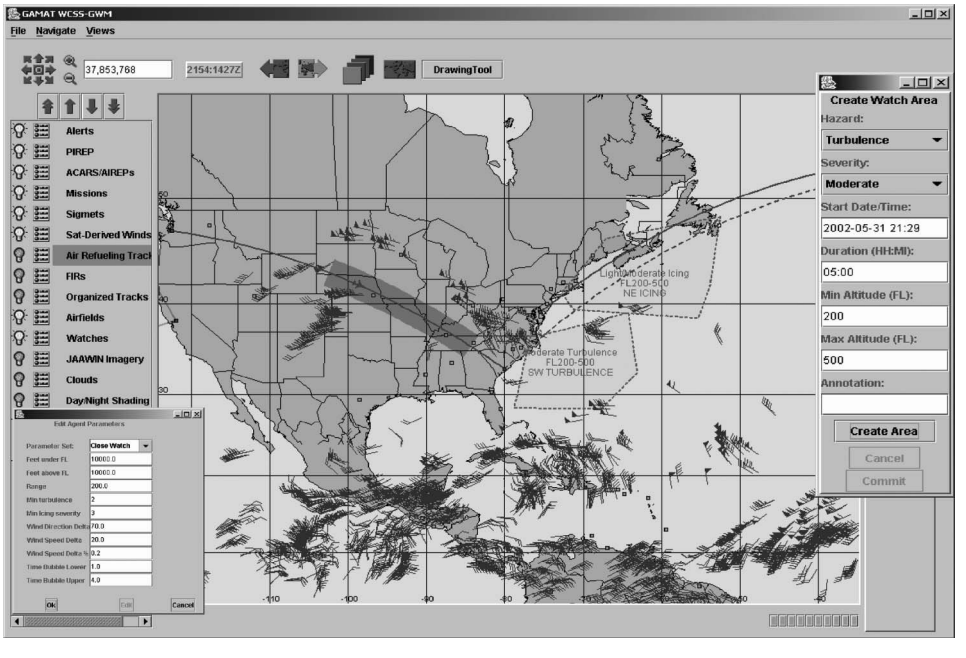

Figure 1. A screen shot from the WCSS-GWM illustrating the ability to create and modify software agents.

3. The need to adapt to a constantly changing world

Over the 4 year time span of the study we observed a wide range of changes that impacted on cognitive and collaborative work in the operations centre. Existing information systems were unable to keep pace with these rapidly changing needs. As a consequence we observed a variety of 'makeshift' strategies and 'home-grown' artefacts emerge as users compensated for the inability of existing software tools to adapt to the continuously changing requirements. Because of the R&D nature of our projects, we were able to make rapid modifications to the WCSS-GWM in response to changing needs. The experience convinced us of the importance of developing software architectures which can accommodate change more readily.

We conducted two analyses to understand the kinds of changes that occurred in the domain and the impacts they had. One analysis examined the kinds of 'work-arounds' and informal artefacts developed by the user community to compensate for the inability of existing software systems to accommodate operational changes. This first analysis looked broadly across a number of different positions within the C2 operations centre. The second analysis focused more specifically on users of the WCSS-GWM. It examined the change requests that were submitted to the WCSS-GWM software design team by the user community in response to changes in work demands. Examination of the results of these two analyses highlighted the need for evolvable work-centred support systems, and pointed to the kinds of capabilities that should be included.

3.1. *Analysis I: user strategies for coping with a changing world*

We conducted field observations and structured interviews in the operations centre over the 4 year span of the project. Field visits occurred approximately every 3 months and were of 2–3 days duration each. Typically, one or two observers sat with operations centre personnel (FMs, weather forecasters, DOs) and recorded the problem-solving and decision-making activities the domain practitioners engaged in, the communications and coordination of activities that occurred with other individuals both inside and outside the control centre in the process of problem-solving and decision-making, how they used existing software tools, including WCSS-GWM, what problems they encountered in using their tools, and the kinds of 'work-arounds' and 'home-grown' artefacts they created to compensate for the limitations of existing systems.

Among the most striking findings from the field observations was the continuous change in the work environment that occurred over the 4 year period of observation. The changes observed included the following:

- changes in goals and priorities of the work (e.g. the nature of flight missions that were conducted, the parts of the world where missions operated)
- changes in the scale of operations
- changes in roles and in team and organizational structure
- changes in experience level of personnel
- changes in the complexity of problems faced (as the number of missions increased the airlift service organization came up against hard resource constraints, making it more important to anticipate and respond to resource bottlenecks and prioritize among missions in cases of goal conflict)

- changes in information sources and information systems provided to support work
- changes in the physical layout of the operations centre (the operations centre was remodelled with the result that forecasters and FM were no longer in as close physical proximity).

While some of the changes were anticipated, others were not. Further, even in the case of anticipated changes, their impact on team roles and work structure were not necessarily foreseeable.

One of the most striking changes was in the scale of operation. As work on the WCSS-GWM programme was starting, in February 2001, the position of flight manager was just being created and staffed. The FMs were only assigned a small percentage of the flights handled by the C2 operations centre of the airlift service organization. Initially, there was an average of three FMs per shift and FMs handled less than 20 flights a month. By February 2004 there was an average of 10 FMs per shift, and FMs handled more than 3000 flights a month.

At the time that the WCSS-GWM was being developed, the organization anticipated, and informed us, that there would be a dramatic increase in the number of missions that would need to be handled and a corresponding increase in staffing. However, while they anticipated an increase in scale, the management of the organization had not determined what changes would be needed in organizational structure to accommodate the increased number of missions. The new organizational structure that was eventually adopted could not have been foreseen in advance.

The increase in scale was accompanied by a shift in team member roles and tasks. While initially a forecaster worked one-to-one with a FM to produce a tailored forecast for each flight-managed mission, the nature of the collaboration between forecaster and FM changed as the number of FMs and flight-managed missions increased. Three separate forecaster positions developed, with one position generating forecasts for different geographic regions, one monitoring 'high-risk' missions, and one responsible for monitoring the remaining lower-risk missions.

The forecaster and FMs now needed support in identifying and managing a set of high-risk missions to focus on, treating these differently from the more routine missions. A series of system change requests were made to allow the WCSS-GWM to import information about 'operational risk management': identifying the high-risk missions and sorting and filtering missions based on risk assessment factors.

Among the consequences of the various changes we observed was a growing mismatch between the support provided by the information systems in place, WCSS-GWM included, and the requirements of the work.

Because the WCSS-GWM was an R&D effort, the development team was in a position to respond rapidly to change requests. In contrast, even simple user change requests to legacy systems required lengthy lead times of the order of months to years to satisfy. The life-cycle of a change request (from prioritization and assignment, through development, test, evaluation, and certification, to deployment) significantly lagged behind the pace at which work demands shifted. As a consequence, we observed users turn to development of informal artefacts, including 'home-grown' software, to compensate for system–work mismatches. Examination of user request changes and informal artefacts which emerged to compensate for rigid systems provided insight into the kinds of change mechanisms an evolvable work-centred system requires to support the evolving nature of work.

Over the course of our field observations we identified a number of cases where C2 staff in the operations centre (FM, weather forecasters, DOs and others) created informal

artefacts to compensate for the limitations and rigidity of existing information systems. These informal artefacts took the following forms.

1. Physical artefacts such as handwritten cheat sheets and sticky notes.
2. New visualizations which graphically depicted important information that was not provided by the information systems as designed.
3. 'Local' databases which stored updates and corrections to information stored in the formal system databases.
4. New software tools programmed by members of the user community to create support systems for aspects of work that were not well supported by the formal information systems.

Physical artefacts generally took the form of hand-written or typed 'cheat sheets' which provided summary reminders of factors that needed to be considered in developing and modifying flight plans (e.g. the location and direction of legal air routes at different times of day). The use of informal physical artefacts is virtually universal across domains, and thus was not surprising to observe (Vicente 1999).

More surprising was the emergence of locally developed software artefacts such as new visualizations, local databases, and 'home-grown' software tools which have not been as widely documented. They point to opportunities to provide more effective work-centred support by providing capabilities to develop these local software artefacts more easily and link them to formally developed work-centred support systems.

A salient example of a new visualization was a case in which users modified an existing timeline display intended to support C2 staff in identifying situations where more planes were scheduled to land at a given airfield than could be accommodated. The display, as designed, focused on showing the number of aircraft scheduled to land at an airfield as the primary indicator of the viability of current landing schedules. However, there were additional important factors that the C2 staff needed to consider that were not visible in the display as designed. These were the operating hours of the airfield, which could change at short notice, and whether the scheduled landing time was during the night or the day since there could be restrictions on whether planes could take off and land during those periods. The users came up with an ingenious way of graphically depicting these important types of information on the airfield displays. They defined 'pseudo-planes' which did not actually exist and scheduled them to be at the airfield during the critical times in question (i.e. when the airfield was supposed to be closed, or when planes were not allowed to fly in or out). By entering these 'pseudo-planes' into the display system, they were able to create graphic visual indicators of information critical to their decision-making that was not anticipated as important in the original system design. This example is similar to examples observed by Vicente *et al*. (2001) of operators creating new indicators and alarms.

The C2 staff also created and maintained local databases that were more accurate and up-to-date than the information stored in the existing formal operational system databases. A limitation of the existing C2 systems that we observed relates to the currency of data. Standard systems are tied to standardized data sources. We observed many instances across different groups within the airlift service organization where the operations staff had knowledge of temporary data changes (unpublicized airfield closures, changes to preferred routing procedures, even late-breaking news of aircraft maintenance delays) which just did not fit into their system. The systems did not provide easy mechanisms for temporary data changes. As a consequence, operations staff could not rely on the validity of the stored data. They resorted instead to developing their own private override databases.

An example is a local database maintained by the operations centre personnel which contained updated base operating hours and temporary closures. While the operational information system contained fields for base operating hours, the information was often out of date. Bases changed their operating hours and declared temporary base closures at short notice. The update cycle for the operational information system databases was not able to keep up with these changes. As a consequence the user community developed its own private local databases.

One of the limitations of these private local databases is that they are difficult to share, even among staff within the operating centre. It was not unusual for many individual controllers to maintain his or her own private database, each containing slightly different information. One of the benefits of creating evolvable work-centred support systems, with explicit provisions for the development and maintenance of local databases, would be the ability to share these local databases across many individuals, fostering shared situation awareness and facilitating collaboration.

The most striking cases of software-based artefacts that we observed were instances where the user community developed its own software tools to support aspects of work that were not well supported by the formal software systems provided and maintained by the larger organization. We observed two clear examples, one developed by the weather forecasting staff and one developed by the C2 staff. In both cases the tools were built by a member of the user community using 'off-the-shelf' spreadsheet and word-processing software. Macros were used to import data from the operational information systems, process and integrate it with locally available information, and then create new displays which better supported the work processes.

In the case of the weather forecasting group, the large increase in missions to be monitored created a need to classify them into different risk-level categories based on a combination of weather-related criteria. There were no provisions in the existing information systems for defining, displaying, or using these risk levels. Consequently, one of the forecasters developed a spreadsheet program to classify and manage missions by risk level.

The C2 staff needed a way of tracking more closely the subset of missions which were considered to be 'high visibility' or which had problems (e.g. missions delayed due to maintenance problems). They created a 'notepad' tool using standard word-processing software with macros that allowed them to import information about these missions from the formal information systems and add detailed annotations as to the current status of the missions and planned actions. Macros allowed the notepads to be periodically updated so that the user could be alerted to new problems. These notepads served as a focused 'to do' list for the users, enabling them to prioritize and manage their work, as well as a shift-turnover log, allowing critical information to be shared across shifts, supporting across-shift coordination and collaboration.

These various examples of 'home-grown' software-based artefacts provide salient examples of the creative 'work-arounds' that users employ to compensate for mismatches between rigid software tools and the evolving demands of work. They point to the importance of developing systems which can be more readily modified by users to support their work.

3.2. *Analysis II: WCSS-GWM change requests*

Examination of change requests to the WCSS-GWM provided a second window into the need for more adaptive systems to keep pace with evolving work requirements.

We identified 50 requests for changes to the WCSS-GWM that were made by the user community. These change requests were classified according to the underlying reason for the change request, and the impact on the supporting software to accomplish the change request.

The aim of the exercise was to understand which change requests resulted from changes in the context of work that could not have been anticipated ahead of time, and to provide a characterization of the types of software changes they entailed. Examination of the kinds of software changes that were motivated by changes in the world provided insight into the kinds of mechanism for change that need to be provided in evolvable work-centred support systems to enable users to adapt the systems to the changing nature of work.

Table 1 summarizes the classification of WCSS-GWM system change requests based on the reason for the request. The majority of system change requests (68%) arose from changes in how the system was used, changes in work processes, organizational changes, changes in systems it communicated with, and other environmental changes.

One of the most common reasons for a change request was expansion of the role of the WCSS-GWM within the organization—either its use by a new category of user, or by expanding the use by an existing user into a new area of work. The WCSS-GWM was originally conceived as a tool to aid collaboration between weather forecaster and flight manager in identifying mission-endangering weather en route. As it came into daily use by both forecasters and flight managers, a number of system change requests were made in order to expand the utility of the system. Most of these changes involved bringing new data into the system and overlaying new information on the maps: air routes, flight

Table 1. Reasons for WCSS–GWM change requests.

Reason for change request	No. of change requests	Comment
New user	11	Additional types of users resulted in expansion of envisaged uses for the aid
New use	6	Original type of user, but expanded scope of use
Unanticipated model of use	3	Original type of user and scope of use (what they would use it for), but unanticipated model of use (e.g. when and how they would use it)
In queue	10	Anticipated functionality on the 'queue' of features to be eventually implemented, or implemented as resources allowed
Environmental change	10	Changes in hardware, software, data availability which impose new constraints or create new opportunities
Uncovering of requirement	2	Uncovering of existing requirement that was not identified earlier (e.g. because of KA sampling limitation)
Change in work process	2	Change in process by which work is conducted
Organizational change	1	Change in structure of organization, change in how work is allocated across individuals and groups
Correction	1	Correction of a system problem
Design improvement	3	Improvement of design based on user feedback/testing
Organizational conflict	1	Reconciliation of disagreement between user organizations

information region and country boundaries, and real-time position reports. All these changes expanded the domain of the WCSS-GWM into areas of the flight manager's job that had not been the target application for the system as designed.

In some instances, weather forecasters made a change request to support unanticipated uses. At one point WCSS-GWM users began faxing maps to flight crew. A request was made to alter some map symbols to make sure that information could be correctly interpreted from a black-and-white fax copy.

Another common reason was environmental change—some externally triggered alteration in data availability, hardware, or software which induced new constraints on or offered new opportunities to the WCSS-GWM. For example, a new weather forecasting software system came on-line for forecasters to use in preparing forecast hazard charts, i.e. maps that identify regions of forecast turbulence or icing hazards. While the new system provided much more detailed weather information, it was not able to overlay flight plans on the same map as weather data. This led to a new requirement for the WCSS-GWM: importing the forecast chart data produced by the new system and overlaying it on the WCSS-GWM map.

The second classification of system change requests was based on the type of software change required. We classified change requests into four broad categories of software impact: data acquisition changes, automated analysis changes, user interface (UI) changes, and software infrastructure changes. The results are shown in table 2.

More than half of the system change requests involved changes to UI. The two most commonly requested UI changes were adding a new type of data to an existing display (e.g. adding air routes or country boundaries) and providing an entirely new functionality in the UI. Newly requested UI functionality could be quite straightforward (adding new sorting and filtering capabilities) or quite complex (provide a new mechanism for reordering map layers). Less common UI changes involved defining entirely new display types, changing the way information is presented in an existing display, or even redesigning the organization of tool palettes.

The second most common software change resulted in automated analysis changes. In our terminology, automated analysis includes any automated processing (rule-based or

Table 2. Software impacts of WCSS–GWM change requests.

Category of software change	No. of changes
Data acquisition	
Acquire new data	9
Change source/format for existing data	3
Automated analysis	
Add new analysis agent	6
Add new processing module for use by an agent or GUI	6
Modify rules of existing analysis agent	1
User interface	
Add new data to existing display	12
Change how data is displayed	3
New type of display	2
New functionality	10
Reorganization of GUI elements	4
Software infrastructure	5

Note: Some changes require more than one category of software change.

otherwise) of information that assists in a decision about what information to display in the UI, how to prioritize information in a display, or how to display information. In the WCSS-GWM system, automated analysis rules are used to alert the user to missions scheduled to fly through forecast hazard areas. Automated analysis rules are also used to colour code airfields, showing airfields operating under visual flight rules in green and airfields with worse flying conditions in a succession of other colours. One of the most common software changes related to automated analysis was to add a new rule-based analysis agent to create a new type of alert. Other typical software changes involved creating new algorithmic procedures to be used by existing agents or graphical user interface (GUI) displays.

Data acquisition software changes typically involved requests to acquire a new type of data (e.g. satellite cloud images, real-time position reports on en route missions, or new forecast hazard charts). In a few cases, other changes needed to be made to data acquisition software to accommodate data format or source changes.

Software infrastructure changes generally resulted from change requests based on system environmental changes. These requests were initiated by new security requirements (replace FTP use by HTTPS) or new network configurations (interact with a newly placed caching proxy server), for example.

Some of the change requests, such as those involving software infrastructure changes, required significant software modification. These types of changes are best handled in the traditional way, with software engineers making the changes to deliver an updated product. However, a significant number of system change requests resulted in changes in automated analysis rules, simple UI changes (adding new data to an existing display), and/or straightforward data acquisition changes (adding a new data source). Enabling users of a WCSS to address these demands for change themselves would dramatically reduce the time required to effect the changes and reduce the need for ad hoc 'work-arounds'.

4. Towards evolvable work-centred support systems

Our observations of the changes that arose in the C2 environment over the 4 year period of the study and the mismatches that resulted between the requirements of work and the software systems in place make salient the need to develop systems that can be more readily adapted to the changing requirements of work. The results also point to some of the kinds of software change mechanisms that would be needed to create evolvable work-centred support systems.

4.1. *Target capabilities of an evolvable work-centred system*

Based on the observations described in section 3, we present a list of 'evolvability capabilities' that characterize an evolvable work-centred system. Each of the items in this list represents one way that such a system could be changed without resorting to bringing in programmers to implement the changes.

- **Bringing new data into the system**. Many of the change requests we saw with the WCSS-GWM were requests to provide new functionality for the user by making new data available to the user. The first step in satisfying such a request is simply to make the new data accessible to the system.
- **Adding new data to an existing display**. Once the data are accessible to the system, they need to be made visible to the user in an appropriate way.

- **Receiving existing data from a new source**. One of the most common change requests resulting from factors outside the control of the users is a change in data source. Whether the existing data are no longer available or there has been a change in format, we need to be able to accommodate such changes easily.
- **Altering the way data are presented in an existing display**. Changing how data are presented in a display (e.g. colour and symbols) is one of the easier types of system change to accommodate. Many existing C2 systems already allow their users to 'customize' their displays in this way.
- **Reviewing and altering the transformation and filtering rules**. The user must be able to understand the impact of the decisions made by these rules. The user must be able to revise the rules to obtain desired system changes and validate resulting system behaviour.
- **Reviewing and altering the behaviour of the presentation module**. The behaviour of the presentation module must be understandable and easily modifiable.
- **Allowing integration with 'homegrown' tools/artefacts**. As users in the operating organization become ever more technically sophisticated, they begin to build spreadsheets and text files that systematize information that is not available in their standard systems. Ideally, our evolvable work-centred systems would end the need for such tools by giving the users enough capability to change the system. At present, it would be prudent to allow easy integration with such user-defined tools by explicitly defining mechanisms for integration with spreadsheets and text documents.
- **Supporting 'local override databases'**. By explicitly allowing for a 'local override database' (a user-controlled database of critical knowledge they have that overrides standard data) our evolvable work-centred systems can make use of the detailed knowledge of the local expert.
- **Supporting test and validation**. Providing test suites to facilitate validation of software changes.

4.2. *Software architecture and technology for building evolvable work-centred systems*

The WCSS-GWM is built on a distributed-agent architecture that has played an important role in our first steps in exploring the capabilities of evolvable work-centred support systems. As implemented, the WCSS-GWM is made up of three main components, each composed of loosely coupled software agents. (The WCSS-GWM is structured as a client–server application: the server contains the data acquisition and analysis modules, and the client contains the presentation module. It is implemented using the D-OMAR distributed agent architecture (Deutsch 1998)).

The first component is a data acquisition module, which is responsible for acquisition, decoding, and storage of data in which any users may have an interest. The second component is an analysis module. In the analysis module, typically some combination of rule-based and algorithmic codes, the raw data acquired by the data acquisition module are filtered and transformed into decision-relevant information of immediate interest to the user. For example, in the WCSS-GWM, the analysis module identifies particular upper-air turbulence observations that threaten the successful operation of air missions. The third component is the presentation module which supports the GUI and provides reasoning algorithms that prioritize information for viewing by the user.

While the WCSS-GWM was not developed with evolvability in mind, the architecture has supported initial steps toward exhibiting a degree of system 'evolvability'.

The following example in particular shows the effectiveness of this architecture of loosely coupled software agents as a basis for building evolvable systems.

The WCSS-GWM had been designed to receive a data feed of composite worldwide satellite images. A single agent in the data acquisition module received these images and the client displayed them as an overlay on the WCSS-GWM map. This worked well until the organization providing the composite worldwide satellite image unexpectedly decided to stop making such images available. The best replacement that could be found was a set of five satellite images which together covered most of the world. By editing configuration files, without changing any compiled software, the WCSS-GWM was reconfigured to accept this change. Instead of a single data acquisition agent, there were now five, each receiving one of the new satellite images. Instead of a single GUI control to turn the satellite image on and off on the client, there were now five GUI controls to control the five separate satellite images. There was even a new menu heading on the client to organize the new set of satellite image controls.

The data access and GUI changes were accomplished in a matter of hours without changing any compiled code. An appropriately trained system administrator could have implemented the changes; the services of a software engineer were not required. This is the kind of user-executed rapid-response capability that we would like to move towards in building evolvable systems.

Based on our experience of building and modifying the WCSS-GWM we can extract certain generalizations with respect to characteristics of software architectures that foster 'evolvability'. The architecture of an evolvable work-centred system must facilitate user-developed additions to system capabilities. This will require an extensive set of user interface components through which a user can accomplish the desired changes. Data access, analysis, and presentation changes will require detailed representations of the relevant domain data objects. For presentation changes, detailed representations of the target display screen entities will also be required. Finally, facilities will be needed to enable users to inspect and modify procedural descriptions.

In our experience a key strategy in allowing new types of data to be easily brought into the system is the development of a structured description of the domain-level data and information objects in the system. Some of these information objects in the WCSS-GWM define air missions, airfields, and weather observations. For each type of object, we currently describe the key attributes of the objects that are used by the system, either for reasoning or for display (e.g. for air missions, key attributes include mission identifier, origin airfield, destination airfield, and schedule). For each type of display provided by the system, we also maintain a structured description of the information objects visible in that display. This description identifies the attributes of the information objects that are primary in the display (visible at all times in the display) as well as those attributes that are secondarily available in the display (perhaps only visible in a mouse-over, or by bringing up a pop-up display). For example, the origin and destination of an air mission are primary attributes displayed by the WCSS-GWM map display, while the schedule for that mission is a secondary attribute available in a mouse-over. These structured descriptions of information and information needs serve as a basis for matching supply and demand for information as system requirements change.

Just as data must be defined so that they can be operated on to accomplish change, there are points at which the process must be specified to accomplish a desired change. Facilities will need to be provided to enable users to inspect and modify procedural descriptions. The WCSS-GWM contains two modules in which process plays an important role: the analysis module uses rule-based and algorithmic techniques to

transform raw data into information pertinent to user decision-making, and the presentation module is responsible for prioritizing information for display to the user. To the extent that the behaviours of these modules can be made rule-governed, there is the potential to enable users to adjust the module behaviours by editing the rules. In this manner, a user could explore and validate desired changes.

5. General discussion

We have advanced the thesis that for a system to remain 'work-centred' over time it must not only support the elements of work identified at a fixed point in time but also include provisions to accommodate change. Our goal is to develop systems that not only allow a user to tailor or customize the interface to meet short-term local requirements but also provide facilities that enable the user community to evolve the entire software structure so as to be able to adapt to the changing demands of the world, i.e. evolvable work-centred support systems. While we recognize the goal as very ambitious, we believe much progress can be made and we have described promising initial design and implementation steps in that direction.

It has long been noted in the human factors literature that users will informally tailor their tools to meet the demands of the work domain more effectively (Vicente 1999). Seminara *et al.* (1977) documented how power plant operators added labels to similar-looking displays and changed knobs on controls to make them easier to tell apart. More recently, Mumaw *et al.* (2000) and Vicente *et al.* (2001) have documented a number of ingenious strategies that operators have developed to compensate for limitations in computer-based information and display systems and make them better suited for support of the work. For example, operators were observed to modify alarm set-points to create new alerts and reminders for action in situations not directly supported by the system as designed. Our findings build and extend on this base. The emergence of locally developed software artefacts such as new visualizations, local databases, and 'home-grown' software tools that we observed in the C2 operations centre is particularly noteworthy as they provide salient examples of the creative, and increasingly sophisticated, 'work-arounds' which users employ to compensate for mismatches between rigid software tools and the changing demands of work. They point to the importance of developing systems that can be more readily modified by users to support their work.

A number of researchers have similarly noted that users will informally tailor the design of their systems and work practices to meet the local demands of the situation better. This has been referred to as 'finishing the design' (Rasmussen *et al.* 1994, Vicente 1999, Mumaw *et al.* 2000, Vicente *et al.* 2001). Vicente (1999) has argued for the importance of creating systems that afford the potential for productive adaptation to enable users to 'finish the design' locally in response to the situated context of work. Our findings and conclusions are consistent with Vicente's proposal. They extend the ideas by emphasizing that the demands of the world are not fixed but will change over time. Thus 'finishing the design' is not merely a matter of responding to specific local conditions but entails adapting systems so as to keep pace with a constantly evolving world; in that sense the design is never really 'finished'.

A question can be raised as to whether some of the requirements for system change that we experienced might have been anticipated if we had performed a more thorough upfront analysis of domain requirements. While upfront analyses are always limited with respect to available time, resources, and access to domain practitioners (Potter *et al.*

2000), and some system requirements were undoubtedly missed by our analysis, that does not fully explain the mismatches that emerged over time between the systems in place and the demands of work in the C2 operations centre. The kinds of forces for change that we observed during our study period produced new requirements which could not have been fully foreseen during upfront analyses, no matter how complete. We contend that, while it is important to ground a system design in upfront analyses of the work domain (Vicente 1999), an upfront analysis, no matter how thoroughly conducted, is unlikely to be sufficient to ensure that a system will display the required flexibility for productive adaptation to future work demands.

A number of researchers have pointed out that when new technology is introduced it can have unanticipated reverberations on the field of practice (Carrol and Rosson 1992, Woods and Dekker 2000). The new system may afford new possibilities recognized by the user community which result in it being used in ways that had not been anticipated by the system designers. Carroll and Rosson (1992) coined the term 'task–artefact cycle' to describe this phenomenon. Clearly the emergence of new unanticipated users and uses for the WCSS-GWM system that we experienced partly exemplifies this 'task–artefact cycle'. However, as we document in section 3, some of the need for adaptation we observed reflected larger forces for change in the work environment.

For a system to remain 'work-centred' over time it must not only support the elements of work identified at the design stage, but must also be able to accommodate elements that the initial design did not appropriately capture and be adaptable to meet the changing nature of the work. While spiral development methodologies attempt to meet these challenges, they have proved less responsive than necessary in the environment in which we have been working and in related environments with which we are familiar. We contend that systems need to explicitly incorporate mechanisms to enable users to adapt the system to evolving requirements.

Our proposal for moving towards evolvable work-centred support systems shares commonalities with recent calls in the computer–human interaction community to move towards end-user development (EUD) systems (Fischer *et al*. 2004, Fischer and Giaccardi, in press). The goal of EUD is to develop tools to enable end-users to adapt and further develop applications to meet evolving requirements. It has it roots in early calls to enable users to create customizations, extensions, and applications so as to address unanticipated requirements (Mackay 1990, Nardi 1993). Fischer and colleagues (Fischer *et al*. 2004, Fischer and Giaccardi, in press) have argued for the importance of developing meta-design approaches which create open systems that can be modified by their users and evolve over time. EUDs range from systems that provide for modest user modifiability to systems that have end-user programming features (e.g. open source codes).

Our findings and conclusions are consistent with a growing recognition in the socio-technical literature that software system requirements should not be viewed as 'fixed' but rather as emergent over time as changes arise in the context of work (Truex *et al*. 1999, Scacchi 2004). Experiences from a variety of sources are challenging the precept that a system can be designed to be correct, consistent, and complete prior to its implementation. Rather, the system development process should be viewed as incremental and ongoing. As Truex *et al*. (1999) have argued, this implies a need for ongoing analysis, negotiated requirements among system stakeholders, and an ongoing investment in software maintenance activities.

The extent to which our users have contributed their own software solutions to meet the demands of their work clearly suggests that users are ready to step in and help address

the demands for more responsive systems. The challenge, as we see it, is to enable the user community to evolve the software structure so as to be able adapt to the changing demands of the world—evolvable work-centred support systems. We believe that such an aim, while technically ambitious, can lead to systems that gracefully evolve to meet changing workplace requirements. This is particularly important for C2 environments which, as we have observed, are constantly adapting to meet changing operational requirements.

Acknowledgements

The WCSS-GWM system development and evolvable systems research documented in this paper was sponsored by the Air Force Research Laboratory Human Effectiveness Directorate.

References

Carroll, J.M. and Rosson, M.B., 1992, Getting around the task–artifact cycle: how to make claims and design by scenario, *ACM Transactions on Information Systems*, **10,** 181–212.

Cherns, A., 1987, Principles of sociotechnical design revisited. *Human Relations*, **40,** 153–162.

Dekker, S. and Woods, D.D., 2000.

Deutsch, S.E., 1998, Interdisciplinary foundations for multiple-task human performance modeling in D-OMAR. In *Proceedings of the 20th Annual Conference of the Cognitive Science Society* (Mahwah, NJ: Erlbaum), pp. 303–308.

Elm, W.C., Potter, S.S., Gualtieri, J.W., Roth, E.M., and Easter, J.R., 2003, Applied cognitive work analysis: a pragmatic methodology for designing revolutionary cognitive affordances. In *Handbook for Cognitive Task Design*, E. Hollnagel (Ed.), pp. 357–382 (London: Erlbaum).

Eggleston, R.G., 2003, Work-centered design: a cognitive engineering approach to system design. In *Proceedings of the Human Factors and Ergonomic Society 47th Annual Meeting*, 13–18 October 2003, Denver, CO (Santa Monica, CA: Human Factors and Ergonomics Society), pp. 263–267.

Eggleston, R.G. and Whitaker, R.D., 2002, Work-centered support system design: using 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).

Eggleston, R.G., Young, M.J., and Whitaker, R.D., 2000, Work-centered support system technology: a new interface client technology for the battlespace infosphere. In *Proceedings of NEACON 2000*, 10–12 October 2000, Dayton, OH.

Eggleston, R.G., Roth, E.M. and Scott, R.A., 2003, A framework for work-centered product evaluation. In *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting* (Santa Monica, CA: Human Factors and Ergonomics Society), pp. 503–507.

Eggleston R.G., Roth, E.M., Whitaker, R.D., and Scott, R., 2005, Conveying work-centered design specifications to the software designer: a retrospective case analysis. *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting* (Santa Monica, CA: Human Factors and Ergonomics Society).

Fischer, G. and Giaccardi, E., in press, Meta-design: a framework for the future of end-user development. In *End User Development—Empowering People to Flexibly Employ Advanced Information and Communication Technology*, H. Lieberman, F. Paterno, and V. Wulf (Eds.) (Dordrecht: Kluwer Academic).

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., and Mehandjiev, N., 2004, Meta-design: a manifesto for end-user development. *Communications of the ACM*, **47,** 33–37.

Kirwan, B. and Ainsworth, L.K. (1992). *A Guide to Task Analysis* (Bristol, PA: Taylor & Francis).

Mackay, W.E., 1990, Users and customizable software: a co-adaptive phenomenon. Dissertation, Sloan School for Management, Massachusetts Institute of Technology.

Militello, L.G. and Hutton, R.J.B., 1998, Applied Cognitive Task Analysis (ACTA): a practitioner's toolkit for understanding cognitive task demands. *Ergonomics*, **41,** 1618–1641.

Mumaw, R.J., Roth, E.M., Vicente, K.J., and Burns, C.M., 2000, There is more to monitoring a nuclear power plant than meets the eye. *Human Factors*, **42,** 36–55.

Nardi, B.A., 1993, *A Small Matter of Programming* (Cambridge, MA: MIT Press).

Potter, S.S., Roth, E.M., Woods, D.D., and Elm, W.C., 2000, Bootstrapping multiple converging cognitive task analysis techniques for system design. In *Cognitive Task Analysis*, J.M. Schraagen, S.F. Chipman and V.L. Shalin (Eds.), pp. 317–340 (Mahwah, NJ: Lawrence Erlbaum Associates, Inc).

RASMUSSEN, J., PEJTERSEN, A.M. and GOODSTEIN, L.P., 1994. *Cognitive Systems Engineering* (New York: Wiley-Interscience).

ROTH, E.M. and PATTERSON, E.S., 2005, Using observational study as a tool for discovery: uncovering cognitive and collaborative demands and adaptive strategies. In *How Professionals Make Decisions*, H. Montgomery, R. Lipshitz, and B. Brehmer (Eds.) (Mahwah, NJ: Erlbaum), pp. 379–393.

SCACCHI, W., 2004, Socio-technical system design. In *The Berkshire Encyclopedia of Human–Computer Interaction*, W.S. Bainbridge (Ed.) (Great Barrington, MA: Berkshire Publishing), pp. 656–659.

SCOTT, R., ROTH, E.M., DEUTSCH, S.E., *et al*. 2002, Using software agents in a work centered support system for weather forecasting and monitoring. In *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting* (Santa Monica, CA: Human Factors and Ergonomics Society), pp. 433–437.

SCOTT, R., ROTH, E.M., DEUTSCH, S.E., *et al*. 2005, Work-centered support systems: a human-centered approach to intelligent system design. *IEEE Intelligent Systems*, **20,** 73–81.

SEMINARA, J.L., GONZALEZ, W.R., and PARSONS, S.O., 1977, Human factors review of nuclear power plant control room designs. Report No. EPRI NP–309, Electric Power Research Institute, Palo Alto, CA.

TRUEX, D., BASKERVILLE, R., and KLEIN, H., 1999, Growing systems in an emergent organization. *Communications ACM*, **42,** 117–123.

VICENTE, K.J., 1999, *Cognitive Work Analysis: Toward Safe, Productive and Healthy Computer-Based Work* (Mahwah, NJ: LEA).

VICENTE, K.J., ROTH, E.M., and MUMAW, R.J., 2001, How do operators monitor a complex, dynamic work domain? The impact of control room technology. *International Journal of Human Computer Studies*, **54,** 831–856. Available online at: http://www.idealibrary.com

WAMPLER, J., WHITAKER, R., ROTH, E., SCOTT, R., STILSON, M. and THOMAS-MEYERS, G., 2005, Cognitive work aids for C2 planning: actionable information to support operational decision making. In *Proceedings of the 10th International Command and Control Research and Technology Symposium*, June 2005. Available online at: http://www.dodccrp.org/events/2005/10th/CD/foreword.htm

WOODS, D.D., 1998, Designs are hypotheses about how artifacts shape cognition and collaboration. *Ergonomics*, **41,** 168–173.

WOODS, D.D. and ROTH, E.M., 1988, Cognitve Systems Engineering. In *Handbook of Human – Computer Interaction*, M. Helander (Ed.) (New York: North Holland, pp. 3–43).

WOODS, D.D. and DEKKER, S., 2000, Anticipating the effects of technological change: a new era of dynamics for human factors. *Theoretical Issues in Ergonomics Science*, **1,** 272–282.