10.20378/irb-58471

User Experience in Technically Complex and Safety Critical Work Domains: Two Case Studies in Aerospace and Aviation

[1]Dorrit Billman, [1]Jessica Lee, [2]Emilie Roth

[1] San José State University &
NASA Ames Research Center, Moffett Field, CA 94035-1000
dorrit.billman@nasa.gov
jessxlee@alumni.stanford.edu

[2]Roth Cognitive Engineering
2 Oliver Court, Menlo Park, CA 94025
emroth@mindspring.com

Abstract. In technically challenging and safety critical work, the primary objective in software design should be supporting execution of that work. Here, experienced usefulness may be the primary contributor to a good "user experience". We report two successful case studies assessing work-needs and designing software; both support planning and scheduling. Case 1 addresses software for planning the movement of the International Space Station (ISS). Case 2 addresses software for planning airlift operations. We compare the processes and representations in the two projects and offer a few suggestions about their benefits.

1 Technical Work Domains and User Experience

We report two case studies for planning and scheduling software in complex, technical work domains. Case 1 addresses software for planning the movement of the International Space Station (ISS). Case 2 addresses software for planning airlift operations. We reflect on commonalities and differences, and what processes and representations might contribute to software that supports user needs in work domains such as these.

In work domains, particularly safety critical and time-urgent work, a key driver of utility is whether workers can use technology to accomplish work goals. In well-designed work domains, while individual and institutional goals may differ, they do not fundamentally conflict. Providing well-designed Human-Technology systems, then, implies providing good support to enable workers to accomplish work goals in a satisfying and satisfactory manner. In this context, then, 'user experience' is primarily oriented to goals originating in the work domain, and outward facing to that external and constrained work; this contrasts with contexts where 'user experience' is primarily oriented to goals originating in personal, individual, or discretionary preference, and thus facing inward toward the user. These contexts produce different forces for allocation of development resources, product design, and marketing strategy.

In our cases, "user experience" was primarily addressed in this work-centered way. Further, in our cases, work was primarily determined at an institutional rather than individual level. There can indeed be disparities between individual and institutional goals; for example, users are often not the voice that drives development or purchase decisions and the view of the decision makers may not accurately capture the work from the perspective of those carrying it out. In addition, the values or measures of remote decision makers may be different from those of users. For example, remote metrics may focus on factors that are more easily measured and "objective" such as time and money, or even margin of safety. Some aspects of effectiveness such as time and accuracy of completing selected tasks can be measured directly and objectively. Other aspects of effectiveness may be much more difficult to measure, such as ability to regulate workload, to recover in unusual situations, to coordinate with entities outside the defined work group, or to discover improved methods of accomplishing work. For these more complex and difficult to measure aspects of effectiveness, User ratings and satisfaction, of both experts and novices, may be a proxy for this type of hard-to-measure aspects of effectiveness. Thus, as has been widely noted (Bisantz & Roth, 2008; Holtzblatt, Wendell, & Wood, 2005), involvement of users in design is beneficial for many reasons ranging from the intrinsic satisfaction it may provide to increased uptake of the developed technology. In our cases, the primary role of users was (with others) to characterize the work to be done, and to provide input on the appropriateness of methods and designs for accomplishing that work.

In short, UE in the context of work, and particularly safety critical work, often is and should be, primarily a matter of effectiveness and safety. Involving subject matter experts to characterize the structure of work is required, and almost surely will include the users of the software being developed. It may also involve other experts and caches of expertise, such as engineers or technical documentation of automation being controlled and other engineered or natural systems in the work domain. Critical knowledge may be distributed across multiple roles, not just users, which was particularly true for our Case 2.

Each case relied on practices oriented to ensure that the systems enabled users to effectively accomplish the work. Neither of the cases reported here relied only on formal requirements to ensure that the developed system was useful, usable, and satisfying. However, both produced documents which collectively guided design and development toward systems that effectively served the users needs. The two cases differed in detail of process and of the representations to support their process.

We present the two cases with particular attention to the role of requirements, documents, and representations. We provide a summary comparison of process and representations used. We offer a speculative analysis of what practices used here will more generally lead to useful and usable system, particularly in agile programming environments.

2 Case 1: Planning the Movements of the International Space Station

The movement of the International Space Station (ISS) is controlled by the Attitude Determination and Control Officer (ADCO) group in NASA Mission Control at Johnson Space Center, together with their Russian counterparts. The ADCO group both builds plans and executes from these plans. An ADCO operator who was trying to improve an ADCO planning tool contacted researchers in the NASA Ames Human Systems Integration (HSI) group for assistance. The AID (Automation-Integration Design) research group at Ames studied a small part of ADCO's work needs, based on limited observation and on analysis of the plans that were products of this group (D. Billman, Feary, Schreckenghost, & Sherry, 2015). This led to a small prototype based on technology built by the Human-Computer Interaction (HCI) group, also within Ames HSI. A laboratory evaluation of the prototype showed large improvements on the selected tasks (D. Billman, Arsintescu, Feary, Lee, & Tiwary, 2015), and this data provided evidence to ADCO managers arguing for funding a larger project. The Ames HCI group had previously built and deployed highly successful planning tools for NASA missions. ADCO was able to secure the services of the HCI group to develop new, production-level planning software. This case study began with initial good will of customers and a technical base in the development group.

2.1 Project context.

Designed to replace multiple legacy planning tools and processes, the ADCO Planning Exchange tool (APEX) incorporates workflow management capability in addition to traditional planning and scheduling functionality. APEX was developed at NASA Ames Research Center by the Human-

Computer Interaction (HCI) group, a group with extensive experience delivering planning and scheduling software for NASA missions. Using contextual inquiry, the team designed and developed the Scheduling and Planning Interface for Exploration (SPIFe), an integrated planning and scheduling toolkit (Aghevli, Bencomo, & McCurdy, 2011). APEX was built by heavily adapting SPIFe to the needs of ADCO users. The project had a two-year timeline with predetermined release milestones and time allocated for testing and transition.

2.2 Team composition.

The team within the HCI group responsible for delivering APEX consisted of one HCI project manager, one HCI product manager who performed user research and design, and a development team of five engineers. From the ADCO group, three end users, all experienced ADCOs at Johnson Space Center, were part of the team: one ADCO project manager, one ADCO project engineer, and one ADCO test engineer. Project managers were responsible for setting the overall timeline, scope, and budget of the project. The responsibility of the HCI-product manager included user research, design, and feature prioritization; this was purposefully intended with the belief that the same individual with deep user and workflow knowledge could most effectively design appropriate solutions and inform feature prioritization with developers. The HCI product manager worked closely together with the ADCO project engineer (who liaised with the broader ADCO team as needed) via weekly phone calls and emails to execute the project. The concentrated amount of knowledge between these two individuals facilitated decision making, communication, and informal requirements gathering.

2.3 Requirements gathering process and work representation.

The foundations of APEX were largely influenced by the integrated planning and scheduling toolkit (SPIFe), previously developed by the HCI group using contextual inquiry and ethnographic field observations. In order to adapt SPIFe to the needs of ADCO users, the product and project manager spent two weeks shadowing ADCOs during their shifts to develop an overall understanding of ADCO workflow and tools. ADCOs, aware of the observers, operated normally and pointed out processes of interest, usually involving software. Notes were recorded in a two-column spreadsheet, with an "observation" column documenting objective observations and an "interpretation" column used for noting implications with respect to product design.

The project managers also provided an initial high-level product requirement document, drawing heavily on required ADCO functions (see Figure 1). For example, requirements included exporting documents with specified information, tracking changes in the plan, and previewing files before sending. Requirements were listed in terms of tool functionality (not design requests) and remained largely stable through the development process. APEX met all requirements, and also provided enhanced automation, visualization, and information organization that significantly improved user experience; the latter was informed by cycles of user observation, testing, and feedback.

Outputs As-Flown TRTL in the following formats
as-flown.trtl
as-flown.trtl.html
as-flown.trtl.plan
As-Flown TRTLs will be modifiable to track events as they were flown
Outputs TCFs per the format defined in OIP [illegible]
Exchanges UAFs, ATLs, TCFs, and TRTLs to appropriate tools and IPs.
Send and receive UAFs to/from MMC via drop-box
Send ATLs to ESA-ATV via drop-box
Send and receive TCFs to/from SUDN via drop-box
Send TRTLs to SACE and PLATO
Ops Timeline, ATL and TCF updates should be completed with minimal user interaction
Changes made to the Ops Timeline that effect an ACR time, will result in the creation of a revised ACR, and updated UAF file for delivery to Moscow.
Revised ACRs will be made available for review prior to sending to Moscow.
After revised ACRs are sent to Moscow, the ATL will be updated with the new ACRs.
Changes made in an ACR time by Moscow will be updated in the Ops Timeline. User will be able to review the change prior to it being implemented in the Ops Timeline.
After ACRs have been implemented in the Ops Timeline, received or approved ACRs will be created for delivery to Moscow. Following delivery to Moscow, the ATL will be updated with the new ACRs.

Figure 1. Part of the initial high-level product requirement document. Requirements were largely driven by institutional and functionality expectations; design specifications (how the requirement should appear and be implemented) was not included.

Due to differences in team location, user research and more detailed requirements gathering were performed remotely via phone and screen share where the ADCO project engineer explained the use cases and details of each requirement. After gaining a basic understanding of the requirements and workflow, the product manager created a document, including a summary of requirements and work representations, outline of use cases, along with several design proposals (see Figures 2 and 3). This document served as an effective communication tool and point of reference for all subsequent conversations relating to the requirement; the ADCO project engineer could provide concrete feedback, discuss advantages and disadvantages of each design, share it with other ADCOs, and point out missing edge cases. After iterations of discussion and design feedback via informal verbal and email conversations, the document provided sufficient detail so that it could be passed on to the developer to implement. Document content varied depending on the requirement; some requirements were more process-oriented (Figure 2), while some requirements focused more on providing visual cues for decision-making (Figure 3).

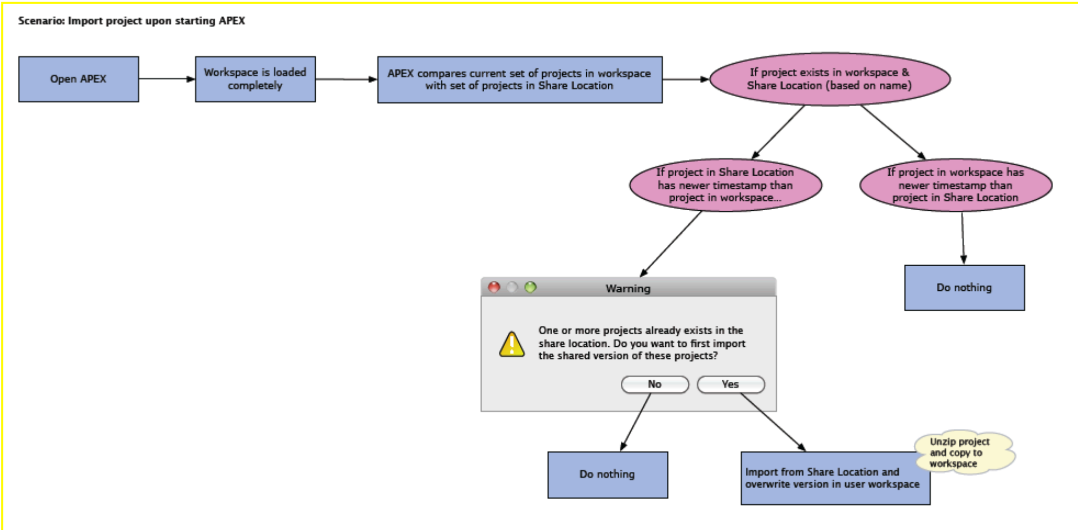

Figure 2. Part of the design document for the file sharing system. The document steps through the process of opening the software, checking against existing files, and providing the user with the option to overwrite.

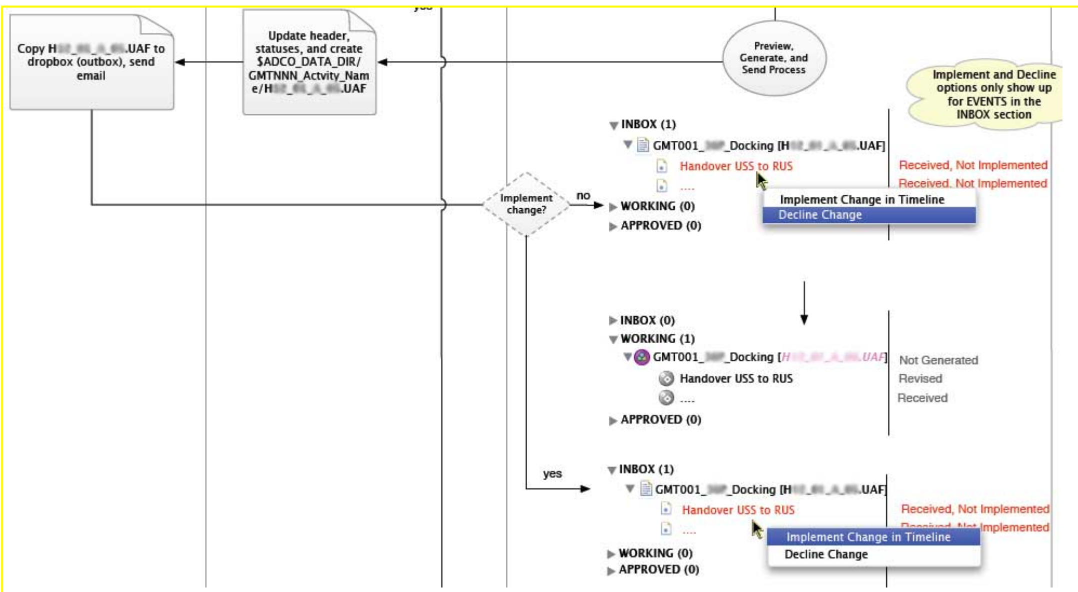

Figure 3. This is part of the design document for the file processing workflow (correlates with highlighted section in Figure 1 requirements document). The left columns indicate the state of the file system, and the right column illustrates the tool interface supporting user decision-making.

Major system-level designs were validated via cognitive walkthroughs using interactive low-fidelity prototypes (PowerPoint with images). After major frameworks were decided, implementation details (e.g. colors, formatting, button placement) were worked out between the product manager and ADCO project engineer. In making decisions, the ADCO project engineer liaised with the ADCO team, and the HCI group actively provided design and usability guidance and input.

2.4 Design, development and evaluation process.

The HCI group worked in 6-8 week development cycles. Prior to the start of the development cycle, the user research and design work already should have been negotiated and completed, allowing developers to accurately assess and estimate fully-developed designs. Several forms of representation were used to communicate the design details from the designer to developers, including workflow showing states of the file systems and corresponding user interface components being affected by the work (Figure 3) and schematics showing the decision structure in scenarios (Figure 2). After feature development, the product manager created testing suites based on the use cases to ensure the acceptability and functionality of the features.

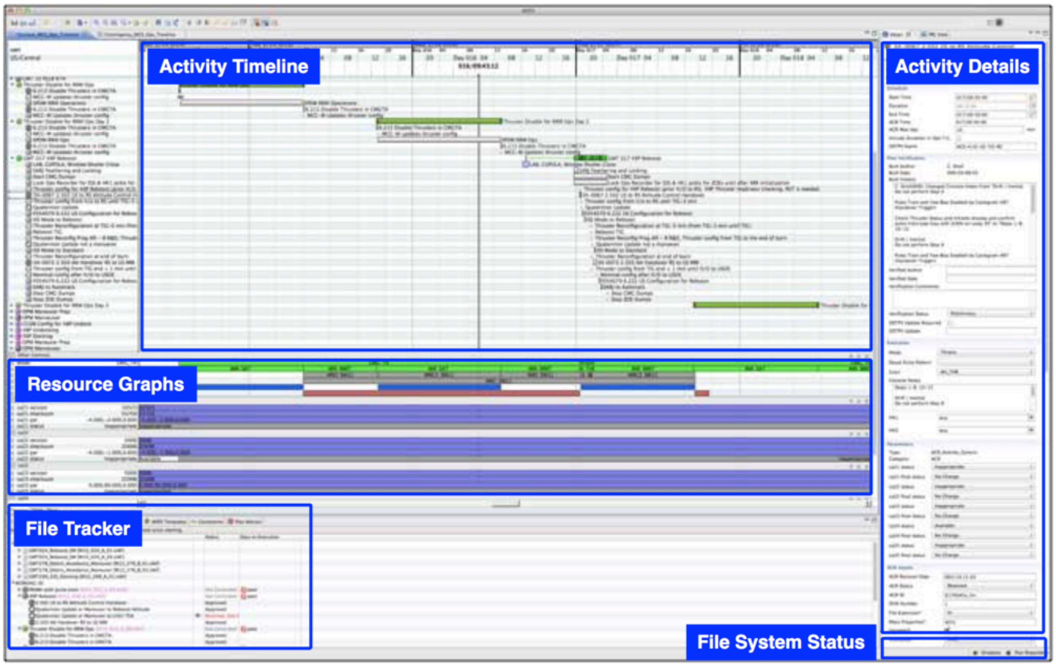

Figure 4. Screenshot of APEX with some major features delineated.

There were seven major releases in the two-year project. It was helpful to provide informal, intermediary releases to give users the opportunity to try out the features and point out problems or usability issues. For each release, documentation of features was included and the ADCO test engineer was responsible for ensuring that the release met criteria. Acceptance criteria included meeting the items as listed in the requirements document as well as user satisfaction with the design and implementation of the features. Figure 4 shows a screen shot of the near fully-developed system.

3 Case 2: Scheduling Airlift Operations

Global airlift operations are complex and dynamic, requiring complex planning and coordination up to and through an operation. The Air Force customer for Case 2 wanted to improve support for scheduling operations and had previously worked with the provider group. The provider group was thus known to the customer, but not to the specific stakeholders in this particular airlift group. Stakeholder trust in and perception of value of the providers' work increased over the project, increasing the providers' access to operators. The provider group had worked together across many years and projects, and had well-established work practices.

3.1 Project context

This case study illustrates work-centered design (WCD). The phrase 'work-centered' is intended to highlight that the focus of the analysis is on the demands and broader context of the work (Eggleston, 2003). The project was conducted for an airlift organization that is responsible for scheduling and tracking airlift missions worldwide. The team developed a decision-support system to enable airlift operations center staff to understand and revise air-lift mission schedules 24 hours prior to and during mission execution (Roth, et al., 2006). This is one of several prototype support-systems this group developed for airlift mission planning and scheduling for the Air Force customer over the span of fifteen years (Roth, et al., in preparation). Typically prototype development projects are one to three years duration, with regular review from an integrated product development team made up of Air Force customer stakeholders. Each project ends with a formal user evaluation, to establish that users concur that the prototype provides effective support for their work, at which point the system is handed over to the Air Force customer for transition.

3.2 Team composition

A key aspect of the WCD approach was that a multi-disciplinary team participated in all phases of the design effort starting from the initial knowledge capture sessions through to the final prototype evaluation. The core team included a cognitive engineer who is able to draw out sources of cognitive complexity and requirements for support through observation and interview of user groups, interface design specialists who are able to translate support requirements into effective graphic visualizations, and software engineers who are best able to grasp technological constraints and possibilities. The dialectic interplay among these multiple perspectives was critical to identifying opportunities for support that were simultaneously grounded in an understanding of what is needed and what is possible, and translating those support requirements into effective visualizations.

3.3 Requirements gathering process and work representation

A fundamental aspect of WCD is an analysis of the work 'context of use' 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 (Bisantz and Roth, 2008) to uncover the characteristics of the work domain, the work requirements, the sources of complexity and cognitive and collaborative demands entailed. Formal methods are then used to represent the results of the analysis. These include work domain representations that capture the goals and constraints of the work and models of workflow within and across individuals and groups required to achieve work goals (see Bisantz and Roth, 2008 for further discussions and examples of work domain and workflow representations).

Figure 5 shows a portion of a work domain representation that was used to capture the factors that need to be considered in understanding and modifying a mission schedule. Work domain representations are used to provide a shared understanding and memory aid of the factors that need to be included in displays intended to aid support mission rescheduling. The representation is used both to communicate inside the design team as well as to communicate to users and stakeholders the current understanding of the design team of the factors that matter. While these representations are developed early in the project, they typically are not needed once the team becomes more versed in the domain.

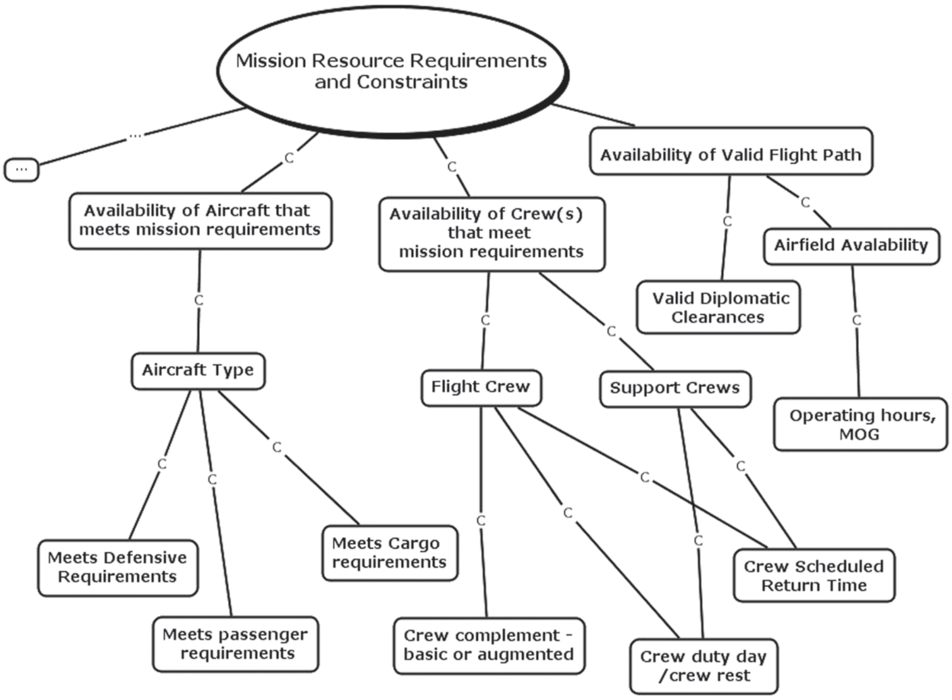

Figure 5. Selected portion of a work domain representation for flight scheduling (Taken from Evenson et al., 2008).

Throughout the development of the work-centered prototype, the design team produced informal products for communicating among themselves regarding the cognitive support requirements that needed to be met and how the evolving design would satisfy them. These included interview notes, notes describing implications for cognitive support requirements, and rapid prototypes of display support concepts.

Once they achieved consensus on the basic design features, they produced materials with which they could present stakeholders the design concepts, their prospective benefits, and scenarios illustrating how the concepts would be employed. A particularly useful intermediate product is a set of 'at a glance' cognitive work requirements that are often stated as questions that the user needs to be able to answer 'at a glance' by looking at the display. Figure 6 provides examples of 'at a glance' cognitive requirements. These cognitive requirements provide the design rational for more specific visual elements and forms that make up the display. These are referred to as 'work-centered display requirements'. Examples are provided in Figure 6. Wampler et al. (2006) provides a more detailed discussion of approaches to communicate cognitive requirements to support traceability from cognitive analysis to display requirements.

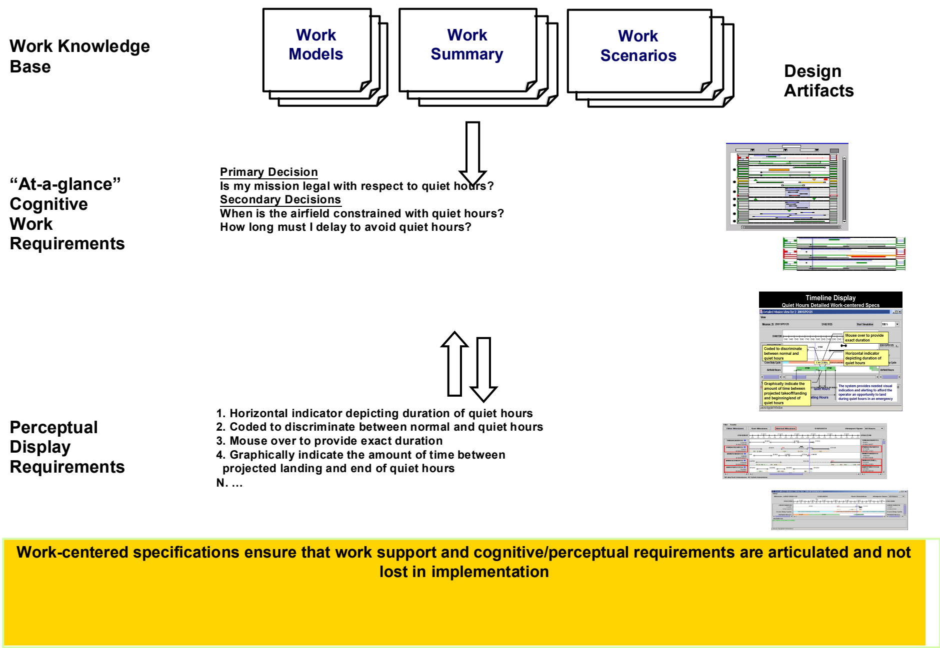

Figure 6. Example of an "at-a-glance" cognitive work requirement. (Taken from Wampler et al., 2006).

3.4 Design and development process

Based on the analysis, a novel timeline display was designed to provide visibility into the domain factors that impact mission viability (See Figure 7). It enables execution personnel to "see" the relationships between mission plan elements (e.g., planned mission take-off and landing times at different airfields) and resource constraints (e.g., airfield operating hours; durations of diplomatic clearances; crew rest requirements). This visualization allows operations center personnel to directly perceive the validity and robustness of a mission plan. Alerts are integrated into the visualization to highlight exceptions and guide problem-solving . In addition an active "what-if" simulation mode is provided to help assess alternative courses of action should mission problems arise (e.g., situations that create mission delays). The user can go into simulation mode and manually 'drag' a mission to a new position on the timeline to see what impact this change has. Alerts come up if the change creates any problems.

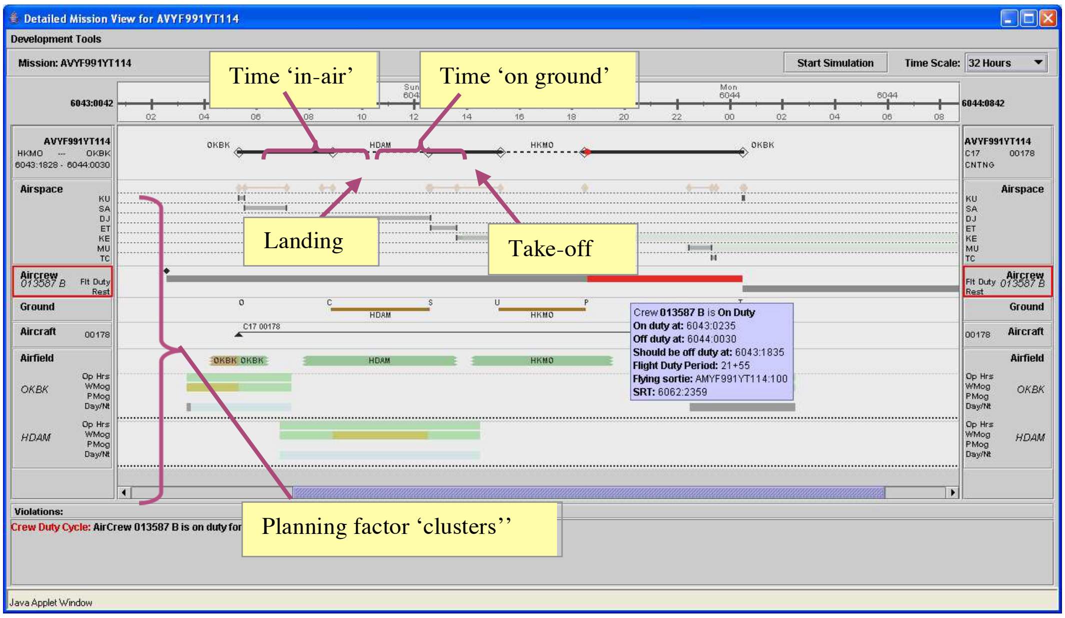

Figure 7. Screen shot of planning resources provided by the software (Taken from Wampler et al., 2006).

3.5 Evaluation and feedback

An evaluation study was conducted to compare user performance with the prototype timeline display vs. performance with the legacy system (Roth et al., 2006). Twelve experienced operations center personnel participated in the study. Test participants worked through different, but comparable, realistic work scenarios in the two test conditions. Results showed statistically significant improvement in terms of both faster solution times and reduced errors with the prototype timeline. Participants took approximately twice as long and made more than twice as many errors in the legacy condition than in the timeline condition. Self-reported ratings of situation awareness and workload were also significantly improved with the timeline vs. the legacy system. The timeline display is currently being transitioned by the airlift organization.

4 Commonalities and Differences

These two case studies were done by unrelated groups, unaware of each other's project, yet had some striking similarities. Concerning the projects, both supported planning and scheduling safety-critical movement, one of an extraordinarily complex vehicle, the ISS, the other of a vehicle fleet. In both cases plans developed over time, and re-planning was an important part of work. In both, plans were affected by technical and political factors, as well as the dynamically adjusted goals to be accomplished. In both cases communication both within the planning group and to other groups affected by the plan were important. Both focused on enabling users to work effectively, safely, and efficiently.

Concerning the processes, both projects had important assets prior to project start. Both were done by or within established working groups, with developed work practices for identifying user needs in difficult work domains and for supporting those needs. Each provider group also had members quite familiar with the type of domain. Both provider groups had established relationships as providers to groups known by and related to the customer, thus increasing initial good will, the expectation of a valuable product, and management support. Both produced successful products. In the Airlift Case, the customer initiated a project to transition the design embodied in the prototype from the research/development group to be conducted by a commercial software development house. In the ISS Case, the implemented design has gone live in ISS Mission Control.

Within this strikingly similar framework, the two projects differed in many details of the processes used and the representations created by and supporting those processes. These are summarized in Tables 1 & 2.

Table 1: Comparison of Process

Processes & Roles	Case 1: ISS	Case 2: Airlift
Provider Roles & scope	1 product manager doing majority of user research and design; 4-5 part-time developers; involved project manager and larger group. Relatively distinct roles and hand-off processes. 2 yr project timeline.	1 cognitive engineer less that half time; 3-4 developers part to full time; additional affiliated SME part time. Highly interactive. Relatively informal and interactive processes. 3 yr project timeline.
Interaction with Users	Deep, on-going relations with one or two user-experts who liaised with user group as needed.	Emphasis on sampling multiple users.
Initial Assets-provider group	Established work practices in larger group	Shared collaborative history of all

	(HCI) but new product manager.	members.
Initial Assets-domain knowledge	Extensive prior experience designing and developing planning software. ADCO work largely unfamiliar.	All members familiar with domain-type and customer organization; previously developed a successfully fielded, work-centred prototype for customer.
Initial Assets-customer relations	Strong reputation known to stakeholders/users.	Strong relation with customer but not stakeholders.
Development structure	Structured Agile development cycles. Product manager completes and negotiates designs prior to hand-off to developers.	Informal Spiral Development. Integrated involvement of multi-disciplinary team throughout.

Table 2: Comparison of Representations

Representations	Case 1: ISS	Case 2: Airlift
Work domain/work needs	Largely informal notes; also indirectly in a) requirements document and b) design document for developers such as how workflow should be displayed.	Initially focal representations of work domain and workflow; less important when team had domain expertise as in this project
Requirements/ project commitments	Formal, stable Requirements specification. Defined scope of contract. Organized sequence of partial releases using agile development	Informal, primarily within group. Group built more formal requirements as a product to guide the developers to whom the project was

	cycle	transitioned.
Solutions, design concepts	Typically proposed multiple designs to discuss with user. Prototypes (interface behaviour in PowerPoint) for major features to confirm design.	Implemented components within the full design are shared to get feedback for one design "spiral"

5 Hypothesized Success Factors and Relevance to Agile Development

The assets at the start of each project were a big contributor to success, particularly development groups with strong work practices, oriented to the user (particularly Case 1) and the work (particularly Case 2). Both projects were moderate scale, but had different overall development structure. Given the similar initial assets, both development structures lead to a successful design.

Many projects will not start with these initial assets. Consider a provider group that has less domain knowledge, less established work practices, or less experience working as a team. Here, what practises or representations can increase the odds of a product that supports the user in carrying out the work, and thus provides a good user experience? We surmise that with developer groups that don't share these assets, the product will benefit a) from explicit and structured interaction within the provider group and between provider and stakeholders and b) with more structured use of explicit, shared, external representations.

The points where explicit representations provide most benefit may vary to some extent depending on where the informal communication channels are most fragile. However, for safety-critical and technically dense work explicit external representations are also very important as cognitive aids for organizing and transforming information, and serve both cognitive and communicative functions. How representation for oneself is related to representation for others, particularly those with less shared context is an interesting and little-explored topic (but see Jamieson, Miller, Ho, & Vicente, 2007). We comment on representations from the perspectives of content topic and of representation use across contexts spanning the needs-analysis through evaluation cycle.

5.1 Types of Representation Content

We group representation types into three very general, higher-level topics and mention examples of each type used in the case studies: representations of the work, representations of the design, and representations of how the design supports the work.

1) Explicit, direct representations of the work were particularly important for both groups in projects prior to the current case studies, when each group was unfamiliar with relevant planning domains. The Case 1 group used Contextual Inquiry and the Case 2 group used Cognitive Work Analysis. In the Case 1 study, the Requirements was an important, stable document and primarily expressed work functions that had to be supported by APEX. Thus, this form of Requirements did much to represent the work domain.

2) Representations of design are particular important if the person designing is 'handing off' the design to programmers, as happened in Case 1. The multi-representation "design document" included specifications showing a work function and how it was to be supported.

3) The relations between design choice(s) and the work function(s) supported by them is valuable information. The relations are particularly important to represent explicitly if a group unrelated to the initial analyst/designer group will be doing the programming. The Case 2 group wrote *requirements* as *output* documents, the set of 'at a glance' representations. The 'at a glance' requirements link cognitive work to a constrained set of allowable design options, that is, those that would preserve the design rationale of the prototype that had been developed by the group. This was critical to protect the cognitive engineering for transitioning the prototype to a production software house.

5.2 Construction and Use of Representations across Contexts

It is very challenging to embody the knowledge resulting from good analysis and design into representations useful for a variety of users and circumstances. For example, often information can best be understood in small components (scenarios, work functions, displays), yet ensuring the coherence of an overall design and its support across the range of intended work functions requires an overall perspective. Representations which facilitate transition from focus on a component to focus on larger or overall structure may be helpful (Billman, Archedeacon, et al., 2015; Pritchett, Kim, & Feigh, 2014) and also visualizations which provide global-local shifts. Ideally, the same content can be provided to guide design and used in evaluation, such that design is guided by the evaluation criteria. Representations most suited for formal methods, as in verification, may not be suited for human inspection, as primarily needed for validation. In addition, the up-front costs of building representations must be balanced against later benefits, often for other people, of having the information explicit and accessible.

Improving the usefulness and hence user experience in technical work domains hinges in large part on understanding the work to be done. The two cases presented here are examples of projects that succeeded, at least in the relative sense of providing much better solutions than legacy software. Both cases developed and depended on effective representations of work needs and of designs to address those needs. Understanding of and development of alternative representations and conditions will be important for reliably producing software for technical work domains that is effective, and thus, produces good user experience.

Acknowledgements

Authors are ordered alphabetically. The authors wish to acknowledge the contributions of JSC ADCO and the Ensemble team, which includes members from the Human-Computer Interaction (HCI) group within the Human Systems Integration Division (Code TH) and the Planning and Scheduling group (Code TI) that contributed to the design, implementation, testing, support, and delivery of the software tool. Part of this work was supported by funding from NASA Human Research Program: Space Human Factors Engineering (466199.02.01). We would also like to acknowledge the interdisciplinary team that contributed to the analysis and design of the airlift scheduling systems.

References

[1] Aghevli, A., Bencomo, A., & McCurdy, M. (2011). Scheduling and Planning Interface for Exploration (SPIFe). Presented at the International Conference on Automated Planning and Scheduling (ICAPS), Freiburg, Germany.

[2] Billman, D., Archedeacon, J., Christopher, B., Deshmukh, R., Ebbs, K., Fan, C., … Stewart, M. (2015). *Needs Assessment and Work Allocation Tools for Mission Operations and Procedures*. NASA Ames Research Center.

[3] Billman, D., Arsintescu, L., Feary, M., Lee, J., & Tiwary, R. (2015). Needs Analysis and Technology Evaluation: Evaluation Case Study of Alternative Software for Controller Planning Work-Part 2. *Journal of Cognitive Engineering and Decision Making*, *9*(2), 186–207. http://doi.org/10.1177/1555343414567775

[4] Billman, D., Feary, M., Schreckenghost, D., & Sherry, L. (2015). Needs Analysis and Technology Alignment Method: A Case Study of Planning Work in an International Space Station Controller Group-Part 1. *Journal of Cognitive Engineering and Decision Making*, *9*(2), 169–185. http://doi.org/10.1177/1555343414567774

[5] Bisantz, A., & Roth, E. M. (2008). Analysis of Cognitive Work. In D. A. Boehm-Davis (Ed.), *Reviews of Human Factors and Ergonomics* (Vol. 3). Santa Monica CA: Human Factors and Ergonomics Society.

[6] Eggleston, R. G. (2003). Work-Centered Design: A Cognitive Engineering Approach to System Design. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *47*(3), 263–267. http://doi.org/10.1177/154193120304700303

[7] Evenson, S., Muller, M. and Roth, E. M. (2008). Capturing the context of use to inform system design. *Journal of Cognitive Engineering and Decision Making*, 2 (3), 181-203.

[8] Gross, T., Gulliksen, J., Oestreicher, L., Palanque, P., Oliveira Patres, R. and Winckler, M., eds. Proceedings of the 12th IFIP TC.13 International Conference on Human-Computer Interaction - INTERACT 2009. Springer-Verlag, Heidelberg, 2009.

[9] Holtzblatt, K., Wendell, J. B., & Wood, S. (2005). *Rapid contextual design : a how-to guide to key techniques for user-centered design*. San Francisco: Elsevier/Morgan Kaufmann. Retrieved from http://www.loc.gov/catdir/toc/els051/2004061522.html http://www.loc.gov/catdir/description/els051/2004061522.html

[10] Jamieson, G. A., Miller, C. A., Ho, W. H., & Vicente, K. J. (2007). Integrating Task- and Work Domain-Based Work Analyses in Ecological Interface Design: A Process Control Case Study. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *37*(6), 887–905. http://doi.org/10.1109/TSMCA.2007.904736

[11] Pritchett, A. R., Kim, S. Y., & Feigh, K. M. (2014). Measuring Human-Automation Function Allocation. *Journal of Cognitive Engineering and Decision Making*, *8*(1), 52–77. http://doi.org/10.1177/1555343413490166

[12] Roth, E. M., DePass, B., Scott, R., Truxler, R., Smith, S. Wampler, J. (in preparation). Designing collaborative planning systems: Putting Joint Cognitive Systems Principles to Practice. In P. Smith and R. R. Hoffman (Eds). Cognitive Systems Engineering: *The Future for a Changing World*. Ashgate Publishing Limited.

[13] Roth, E. M., Stilson, M., Scott, R., Whitaker, R., Kazmierczak, T., Thomas-Meyers, G. and Wampler, J. (2006). Work-centered design and evaluation of a C2 Visualization Aid. *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*. (pp. 255- 259). Santa Monica, CA: Human Factors and Ergonomics Society.

[14] Wampler, J., Roth, E., Whitaker, R., Kendall, C., Stilson, M., Thomas-Meyers, G., and Scott, R. (2006). Using work-centered specifications to integrate cognitive requirements into software development. *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*. (pp.240- 244). Santa Monica, CA: Human Factors and Ergonomics Society.