# Needs Analysis: The Case of Flexible Constraints and Mutable Boundaries

**Dorrit Billman**
San Jose State University &
NASA Ames Research Center
Moffett Field, CA 94035-1000 USA
dorrit.billman@nasa.gov

**Michael Feary**
NASA Ames Research Center
Moffett Field, CA 94035-1000 USA
michael.s.feary @nasa.gov

**Debra Schreckengost**
TRACLabs
1012 Hercules
Houston, TX 77058 USA
ghost@ieee.org

**Lance Sherry**
Center for Air Transportation Systems Research
Fairfax, VA 22030 USA
lsherry@gmu.edu


*CHI 2010*, April 10-15, 2010, Atlanta, Georgia, USA
ACM 978-1-60558-930-5/10/04.

## Abstract
Needs analysis is a prerequisite to effective design, but typically is difficult and time consuming. We applied and extended our methods and tools in a case study helping a mission control group for the International Space Station. This domain illustrates the challenges of information-system domains that lack rigid, immutable, physical constraints and boundaries. We report the successes & challenges of our approach and characterize the situations where it should prove useful.

## Keywords
Needs Analysis; User Studies; Planning; Cognitive Engineering

## ACM Classification Keywords
H1.2 User/Machine Systems (e.g., HCI)

## General Terms
*Human Factors*

## Supporting needs analysis
Our group develops tools and methods to support system design and analysis. We focus on systems that have complex interaction and allocation of functions between human and computer (e.g., automation). Needs analysis is an important precursor to design

specification. It is the process of identifying what functionality is needed in a new or revised computer system, to guide design of the system and the accompanying interaction.

Needs analysis may identify constraints in the domain of work or activities to be carried out. Domain constraints include physical limits or requirements on a system being controlled, such as how quickly a vehicle can be moved from one position to another. Activities or tasks are the goal-directed behaviors done to bring about some desired state, outcome, or product, and they can be described at a more specific or more abstract level. Needs analysis can be difficult, time consuming, and costly, but if not done well, the new system may solve the wrong problem (Leveson, 1995). Our research goal is to develop tools and methods to help domain experts provide the information needed for design, efficiently and effectively.

To develop tools and methods, we need example domains. We found one such example domain in the Attitude Determination and Control Officer group (ADCO), a group within the National Aeronautics and Space Administration (NASA) Mission Control for the International Space Station, in Houston. Broadly speaking, ADCO flies the International Space Station (ISS). ADCO flight controllers are responsible for determining and changing the orientation and trajectory of the ISS. Most frequently, they change the way the ISS is oriented in its orbit but they also are responsible for reboosts to move the ISS back up into a higher orbit. They work in close collaboration with Russian counterparts, as the mechanisms for changing position and attitude are split between Russian and American systems. They also draw on intensive engineering support for modeling and deriving technical specifications.

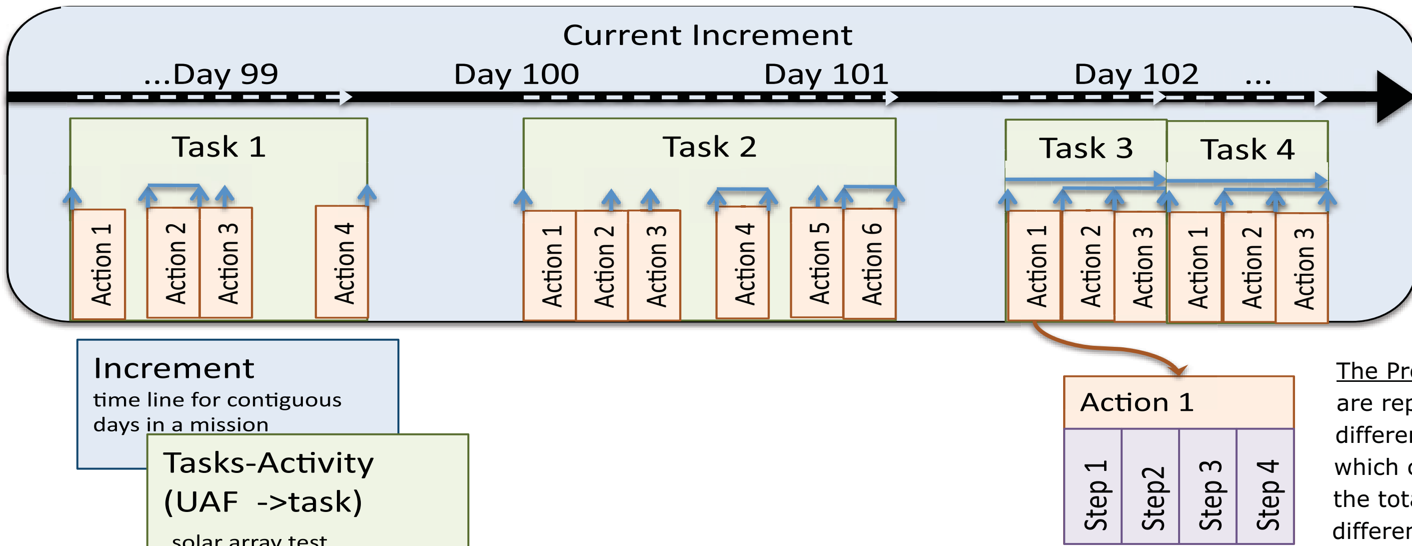
ADCO flight controllers had concerns about the computer systems they use in planning. In particular, a key tool used to exchange developing plans with the Russians was a text editor with formatting support, and this was experienced as very inefficient, and error-inducing. This gave us a chance to study the group's planning needs and to apply and refine our tools based on what we discovered we needed for this analysis.

### A Case Study on Two Levels.

This Case Study operates on two levels. The primary level concerns tools and methods. The secondary level concerns our example domain, ADCO, to which we applied the evolving tools and methods.

For clarity, we begin with the concrete: our secondary-level results, the key findings from our ADCO planning needs-analysis, using the representations we developed over the course of our research. Though we present this as a 'matter of fact' description of ADCO planning, none of this characterization was available prior to our work. Documentation of ADCO planning tasks is minimal, though instructions for using the existing planning tool are given. After presenting these secondary-level results, we summarize our historical method used to arrive at our needs analysis, or how we know what we know.

We step back to summarize what we learned from this process concerning our "primary level" results, namely, the evolving tools and methods we think valuable for doing needs analysis. Following this, we share our approach to the broader principles for needs analysis



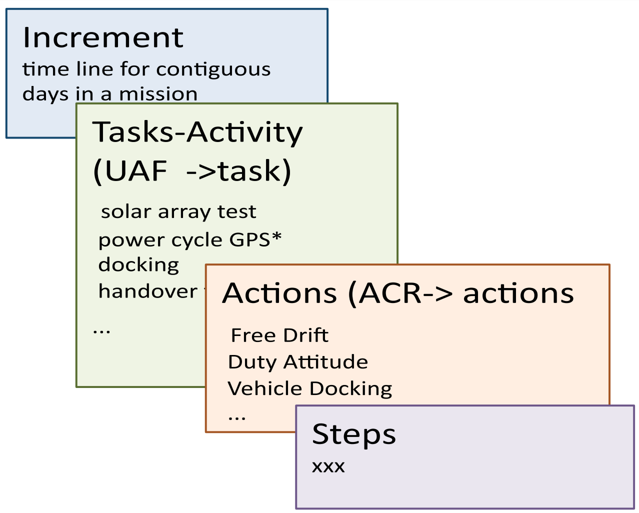




**figure 1** Structural constraints on product: the temporal and part-whole composition of ADCO plans. (A plan is developed by ADCO and later used to guide execution of ISS maneuvers).

and relate our approach to other work. We summarize the lessons learned in terms of when to use our approach. Then we describe the intended beneficiaries of our work, and its limitations, before concluding.

### Secondary Level Results: Summary of ADCO Planning Needs

ADCO's Planning. ADCO flight controllers do extensive planning, collaborating internally and with other groups. Our needs analysis addressed only (part of) the planning activities, not the real-time monitoring and control of flight. ADCO acts as integrator and coordinator for much of the planning activity, which requires input from and notifications to multiple groups.

The Product: A Plan. ADCO plans are represented in several different files and documents, which contain different parts of the total plan-information, are differently formatted, and meet the needs of different groups. However, explicit identification of the underlying informational structure and content of ADCO plans, was a critical part of the needs analysis.

Plans are organized in a part-whole structure spanning intervals and points in a time-line, illustrated in Figure 1. One plan spans an Increment (typically a period of months bounded by change in ISS crew). An increment includes a sequence of Tasks involving ADCO, such as vehicle docking and scientific testing. For any ADCO Task, there are a series of actions, which primarily involve changes among control mechanisms and use of the active control mechanism to change the attitude of the ISS. Each of these three components is planned in great detail and with extensive collaboration. Each Action has about 16 attributes. These specify engineering parameters (e.g., the yaw, pitch, and roll of a target attitude), formal approval status of the action, and communication or metadata tracking (e.g., who specified the action and when). Steps specify the procedures used to carry out the plan, are internal to ADCO, and, because they are heavily standardized, are more critical for execution than for planning. (The part-whole time-line representation was developed and filled out by a human factors expert and vetted by domain experts.)

Our analysis focused on the document passed back and forth between US attitude planners (ADCO) and the Russian attitude planners, a text file called a UAF, or Unified Attitude-change-request File. A UAF file typically represents a Task, and lists the sequence of Attitude Change Requests (ACR) specifying the attribute values for each component Action.

The Process of Plan-Building: Decision-Level Tasks. What are the activities that ADCO controllers do in building a plan? Overall, planning requires checking technical specifications for a proposed maneuver, including things such as whether the proposed attitudes have been validated by engineering analysis (e.g., for balancing atmospheric drag and solar wind, for minimizing fuel usage), or whether the inertial mass-specifications match the current ISS configuration. Planning also requires consulting with other US flight disciplines (e.g., station electrical power) and with Russian counterparts about the effects of attitude changes on other systems (e.g., power generation) and human work systems (e.g., scheduling a maneuver at a time when it can be appropriately supported by multi-time-zone work groups). Steps were listed at the level of cognitively demanding decisions, not at the level of "button presses," nor specified in terms of current tools. These describe the task in terms of its purpose, specify what triggers need for the task, what information is needed to do it, what output results, and points to any subtasks of sufficient complexity to need further description themselves. Figure 2 shows an example of an ADCO planning task. The *Check TEA attitudes* task is part of *Reviewing a UAF* (shown in Figure 3) and requires consulting documents or phoning another controller group to verify planned values specified in the UAF (Unified Activity File), used to communicate with Russian counterparts while planning.

**figure 2.** An example decision-level task, shown in one row of the Mission Decomposition Matrix. The *Check TEA attitudes* task is a lowest level task but is still specified abstractly, not at the level of "button pressing."

**Context Task= Review UAF**

| Task | Freq. | Trigger | Function<br>*Sources of Information* | Task Output | Subtasks |
|---|---|---|---|---|---|
| Check TEA attitudes | As often as an UAF arrives | read UAF | Verify attitudes labeled TEAs are correct.<br>*• Determine vehicle config with diagram under the glass.*<br>*• Determine solar array config using sola array plan, current telemetry or Phalcon call*<br>*• Determine beta angle with baseplate* | possible change to ACR | |

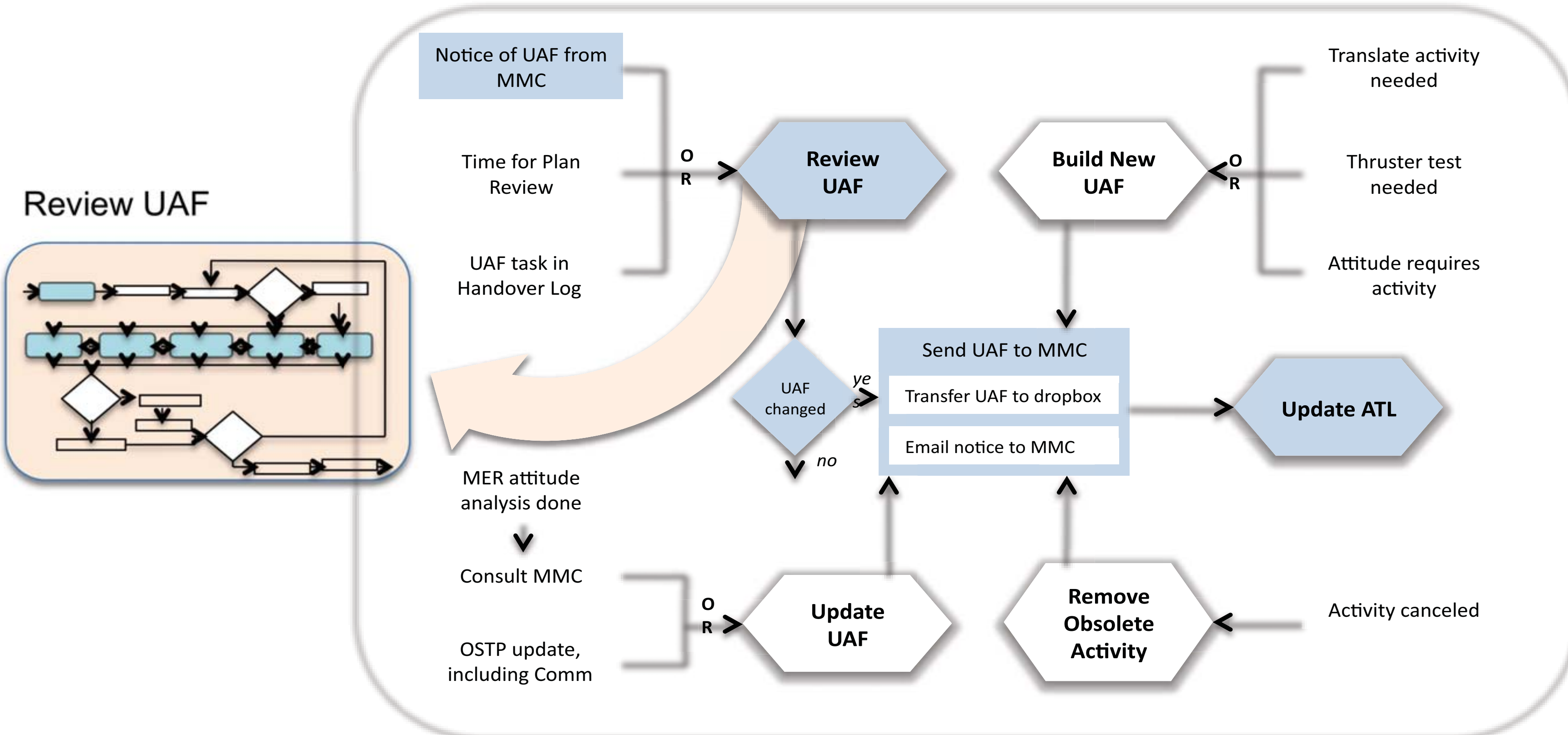



**figure 3.** The main frame shows the highest level flow and control, in the task decomposition diagram; blue (shading) highlights one possible sequence of actions. The inset frame indicates the diagram decomposing one of the high level tasks; aqua (shading) indicates a task with further component

<u>The Process of Plan-Building: Communication, Information Flow, and Overall Task Structure</u>. Communication requirements, both informal exchanges and formal approval, provide the high-level organization of planning activity. Figure 3 shows the top-level task structure, which is primarily communication. High-level tasks are expanded into the lower level activity, as illustrated in the figure inset.

| CONSTRAINTS: PRODUCT [Engineering] |
| --- |
| ADCO reduces the threat from violating any of the 4 constraint types<br>-by detecting and correcting unintended values specified in a plan (e.g., through data entry errors)<br>- by identifying situations when the intended values violate constraints (e.g., because a problem has developed that puts constraints in conflict) and participating in problem solving to avoid constraint violation or minimize adverse impact where this is not possible. |

| CONSTRAINTS: PROCESS [time] |
| --- |
| ADCO planning process must meet timetable requirements for when different types of information need to be correctly specified.<br>These requirements are "soft", because there can be reasons why "late" modifications to a plan are necessary. |

**figure 4.** Example constraints from the Constraint Agreement & Importance form, illustrating the negotiated, flexible aspect of both process (plan building) and content (plan) constraints.

Constraints on Plans and Plan-Building. ADCO planning needs can also be characterized in terms of constraints or requirements on both product and process. Constraints on the product include the part-whole structural constraints described above, requirements on engineering relations, and on formal approval status. Constraints on the process include specific timing requirements and broader requirements for building social capital to maintain working relations among groups. Figure 4 shows examples.

**Secondary Level: The Story of Our Analysis**

The ADCO group wanted to redesign tools used in planning ISS attitude and had begun specifying improvements for the interface. A critical "pain point" in working with the existing system was exchanging information with counterparts in Moscow Mission Control to develop the joint plan for future ISS maneuvers. Specially formatted files (UAFs, or Unified Activity Files) are passed back and forth as refinements or revisions are made and each action in the UAF goes through a multi-step approval process. Currently, controllers use a form editor to carry out this planning and scheduling activity. We urged identifying needs before redesigning the tool.

| | Focus of Activities |
| --- | --- |
| Phase 1 | Orientation & Definition |
| Phase 2 | Low level task analysis |
| Phase 3 | Alternative Considerations and Mission Decomposition Matrix |
| Phase 4 | Complementary Representations: Showing relations among tasks |
| Phase 5 | Widening the Boundaries and Focusing on Constraints |

We report the story of our analysis process, conducted jointly with ADCO, in 5 phases. We describe the motivation for the representation tools we developed along the way, partially illustrated in Figures 1-4.

Phase 1: Orientation and Definition. The ADCO expert had seen a presentation on some of our methods and thought a task analysis tool might be helpful. Initial discussion addressed how our human factors research group (HF) and this domain expert (E) could work together as well as an introduction to the domain and to the goals for tool improvement. The ADCO expert had identified the UAF editor (used to build plans exchanged with the Russian counterparts) as the element most in need of a redesign. We set the analysis boundaries to the task of editing and exchanging these files, thus setting the scope of analysis to considering the constraints, goals, tools, or tasks concerned with this aspect of plan development. HF used a form of contextual inquiry (Beyer and Holtzblatt, 1998) by observing E on-the-job and questioning E about the job. In consequence, HF begins to understand the broad nature of the planning activities.

Phase 2: Low-level task analysis. We provided ADCO with a lightweight, web-based tool for task analysis, eHCIPA (electronic Human Computer Interaction Process Analysis, Sherry, Fennel, Feary, & Polson, 2006). It had been designed for an unaided domain expert to evaluate usability. It asked the user to document tasks with their associated low level actions and rate how transparently each was supported by the interface; using this information, it provides feedback about usability of a design. This tool was not compatible with this stage of analysis. Each low-level task had to be specified as an individual components, but it was difficult for E to segment these from the overall work flow. Further, E had internalized complex sequences of actions and found it tedious to specify the "obvious" components, particularly when they had to be entered repeatedly for different tasks. These difficulties were compounded because the tool did not provide hierarchy nor guidance about the level of task to enter.

By focusing on low-level tasks, the task description was most dependent on the details of the current method and system. Thus, the very aspects of the task that were the focus of the description were those most likely to change. As a result, much effort went into building parts of a task description probably irrelevant to the new system. In sum, though eHCIPA is useful supporting evaluation, its focus on low level details made it inappropriate for supporting this early phase of needs analysis. Although Phase 1 had included some high-level task description, we shifted too quickly to a very detailed level.

Phase 3: Alternative Considerations and Mission Decomposition Matrix. We shifted granularity up to larger-scale, mission-level tasks. We developed a new tool, called the Mission Decomposition Matrix (MDM) shown in Figure 2; it evolved from the Task Design Document (Sherry & Feary,2004), as had eHCIPA. Like eHCIPA, the MDM remained task-oriented, capitalizing on the fact that tasks or activities are fairly accessible and can be reported fairly accurately; this is particularly true for tasks that require deliberation rather than automatic response. The critical change was to request report of tasks or functions at a higher, abstract *mission* level. We use the term mission to emphasis that mission-level tasks drive the analysis. Rows in the matrix represent an individual task or mission, and columns represented aspects of that task or mission (see Figure 2). The key columns specify an informal name given to the task (Col 1), a description of the function of the task and the sources of information needed to carry it out (Col 4a&b), and the product or result of completing the task (Col 5).

Although a high-level mission decomposes into its constituent activities or tasks, the "initial draft" of the MDM did not represent these relationships between tasks. E added hierarchical structure to the matrix representation in two ways. First, he added sub-matrices for each subtask, rather than listing all tasks in one matrix. Second, he added a column for each row listing any subtasks for that task. Now, the decomposition of a mission into subtasks or component missions was explicit. Further, we added the category *sources of information*, over the course of using MDM, which was jointly completed by E and HF. (The matrix representation was developed jointly by domain and human factors experts, filled out by the domain expert, elaborated by a human factors expert, and checked by the domain expert.)

The extended, completed MDM provided a good characterization of the individual tasks done to accomplish the work functions, provided information about the input and output of each task, and flagged the importance of hierarchical structure. However, the matrix representation did not do a good job representing relations among tasks.

Phase 4: <u>Complementary Representations: Showing relations among tasks.</u> To provide a complementary representation that captures the relations among tasks, HF represented all the task relations in a task decomposition graph. Each graph shows tasks at a single level of abstraction, but higher-level tasks were expanded into their subtasks in separate graphs. Figure 3 shows the flow among top-level tasks. Any (sub)task was expanded only once even if used in multiple tasks. HF used the information from the MDM to construct the structure of the graphs. The decision structure among tasks was added.

These graphs were reviewed by E and by a second ADCO user, working separately. The graph representation elicited some additional information and also prompted slight modification to structure, beyond what E had listed in the MDM. HF revised the graphs as specified and E reviewed these.

The task decomposition diagram used initially was developed by the HF group. We are investigating the use of the Unified Modeling Language (UML; http://www.uml.org/) to represent the task decomposition information, specifically the UML activity diagram. We have successfully migrated a portion of the task decomposition diagrams to UML at the time of publication.

Phase 5: <u>Widening the Boundaries and Focusing on Constraints.</u> The task analysis done with the MDM lead to a better understanding of the activity around editing the UAF, exchanging information with the Russians, and general attitude planning. However, it became clear that focusing on the UAF editor was too narrow. There were several, critical high-level functions in which scheduling and communication played a very prominent role. Many of the time-consuming and thought-demanding activities were difficult because of scheduling and communication needs rather than difficult engineering. Many tasks were organized around scheduling, suggesting that the UAF editor was not the best tool for the job, and that tasks might be eliminated and simplified with a tool that supported scheduling aspects of the task. We identified and demonstrated a new NASA scheduling tool (McCurdy, 2009), as a possible prototype better adapted to ADCO's needs than the text editing tools in use. (See also Butler et al, 2007 on design for scheduling.)

For this broader problem, we wanted to explicitly represent the demands and constraints, which we believed were implicit in the information gathered so far (existing documentation including the Station Console Procedures, SCP; MDM; task decomposition graph; and the hours of interaction with the expert). We drafted a mission statement and identified four quite varied types of constraints, concerning the product or process.

A focus on constraints is shared with Cognitive Work Analysis (CWA) methods. Its advocates have also noted the cost of (low-level) task analyses. However, they stress the importance of *first* identifying the constraints from the domain that restrict viable actions rather than focusing on the activities themselves. Its advocates claim CWA will be useful in design of complex, safety-critical, dynamic socio-technical systems. This characterization fits the ISS and its supporting systems. However, for our task (despite fitting this characterization) it would have been difficult to do the constraint analysis first. The high-level task analysis elicited from experts was a prime source for defining boundaries and identifying constraints.

Most fundamentally, we provided an explicit characterization of the structure of the product: specifically, an ADCO plan is a temporally ordered part-whole activity hierarchy. The plan structure was represented graphically, shown in Figure 1. In addition, the engineering content of a plan must meet engineering constraints. For example, engineering constraints included physical, control, circumstantial, and asset conservation. Third, a plan must comply with format requirements. (While the specifics of format might change, adherence to some fixed output format is required to enable reliable communication among groups. Considerable effort is spent ensuring format requirements are met.) The fourth type of constraint concerned process, ranging from requirements that plan components are approved, that information needs to be specified in time to meet known planning deadlines, or that general practices to cultivate good will among groups are followed.

Each constraint was framed as a requirement or something ADCO should accomplish, for example, what should be true of a plan, or how ADCO must interact with a "customer" group. We tried to capture high level, difficult, and relatively stable constraints, rather than listing, say, the current formatting requirements.

One striking feature of our constraint set is that many of the important constraints in this domain are "soft," or negotiable: while flight rules are important, there can be exceptions; while deadlines are important, they may need to slip.

All functions were listed in a form to be completed in a structured interview. Example entries are shown in Figure 4 (though the response options are not included). This structure was elaborated by an expert, who added the *Steps* level that links to the formal procedures for executing the step; this level is clearly important in execution, and is part of the plan, but plays a limited role in planning. Three expert users (including E) were interviewed individually. All the functions were vetted as accurate. Some rewording was proposed and elaborations and explanations offered.

Summary. As a result of these five phases of activity, we have a characterization of the high level tasks and the critical requirements and constraints of the planning mission of the ADCO group. This can be translated into a design oriented needs analysis.

## Primary Level Results: Recommendations about Needs Analysis

Our goal is to provide effective and practical needs analyses, to guide design of interaction and of the systems behind the interaction. We make two broad claims.

First, we believe that a needs analysis should generate multiple, external, sharable representations. These should 1) support communication and negotiation, 2) support crosschecking and problem finding through comparison of alternative formats, and 3) make explicit claims about task structure and about domain structure, which can drive design. We expect that construction of multiple, differently structured representations will prove the most efficient path to good design, rather than adherence to a single primary format (e.g. an Abstraction Hierarchy, GOMS description), however valuable it may be. We found high-level, hierarchical task listing (MDM); task decomposition diagrams; and descriptions of (soft) constraints all valuable and complementary. We think the process of analysis was speeded by shifting among these.

Second, we claim that both high-level activities and domain constraints are important parts of needs analysis. Further, we recommend starting with task analysis, and then flexibly iterating between identification 1) of *high-level* tasks and 2) of domain constraints.

This contrasts with many task analysis methods that emphasize detailed tracing of specific actions (e.g. many in Diaper & Stanton, 2004 including GOMS-based methods). Such task analysis methods, while suited to evaluating existing systems, do not provide the best guide for designing novel systems: the action details that are the focus of analysis are just the aspect most likely to be irrelevant, as old actions are transformed and new ones introduced. Our approach is more consonant with high-level GOMS analysis (Kieras, 2004), which seeks to guide design by focusing on a top-level decomposition of goals. The advantage of a single unitary representation is that it can be “unpacked” down to the level of completely specified actions; its cost may be difficulty in generation or ease of missing information (constraints) that might be visible in an alternative representation format. It relies on the ability to specify some procedure sufficient to accomplish the goal and in specifying a particular strategy, may be overly specific (e.g., assuming a chip design is constructed by drawing rather than by recomposing old cases). However, focusing on procedures over constraints may not be a cost in domains where the challenge is discovering any method sufficient to accomplish the task.

Our position shares similarities to the ecological approach. In particular, our content claims—that analysis of activity (control task) and of domain are both important for design-- are in accord with claims from ecological interface design (EDI) (Burns & Hajdukiewicz, 2004) and cognitive work analysis (CWA) (Vincente, 1999). Dowell's (1998) recommends analysis in terms of *domain*, in combination with analysis of *worksystem*. In addition, ecological advocates have also pointed out the low relevance of carrying out a low-level task analysis on the old system. Further, we share with them the goal of designing to support and elicit expert performance.

However, consider our claim about process--that starting with a high-level activity analysis then iterating between activity and domain analysis is an efficient and effective method for needs analysis. This runs counter to the recommendations of CWA and EID. These approaches have strongly proposed a logical and methodological ordering: 1) boundaries of the system to be analyzed must be set first, 2) domain constraints must be identified next, and 3) control tasks analyzed afterward. We believe this would not have worked well in our circumstances. We next characterize when to prioritize analysis of task over domain constraints and to use iteration over linear sequence.

## Primary Level Results: When to use our approach--Flexible Constraints and Mutable Boundaries

Our domain includes a physical system being controlled (the ISS); tremendously safety critical operations; dynamic, time-pressured aspects of work; and a highly complex socio-technical system. This description matches the "heartland" of applications where ecological cognitive engineering has been applied. Yet our situation has several properties that suggest a process that a) is iterative and b) uses high-level task analysis prior to and as an entré to capturing domain constraints.

1) System Scope: Mutable Boundaries.
The boundaries of the system are subject to change both because the system itself changes and because the analysist changes the boundary definitions. The initial system boundary, per the ADCO user request, was an editing tool and the functionality it supported. As many before us, we discovered that initial user characterizations, of problem as well as solution, may not be correct. Rather, through the collaborative process of task analysis, we identified a much broader scope that should be included in the needs analysis. The initial system scope was important, because it identified the core of user concern and motivation, but it was a starting point, not a good system boundary to retain. A better boundary emerged over the course of task analysis.

2) Physical system: Unknown constraints and flexible resolution.
The ISS certainly has physical limits and any plan should specify actions within these limits. However, the actual limits of performance of the ISS are incompletely pre-specified because system configuration is highly dynamic. The ISS systems are unique and complex. The subsystems are highly coupled; each subsystem is intensively modeled but understanding of the whole, by anyone or any group, is incomplete. This incomplete understanding, with accompanying risk, is recognized and is accommodated by practices designed to keep operation well within regions known to be safe. The practical goal is to keep operations as centrally located

within known safe regions as feasible, while still carrying out a complex mix of goals, rather than excluding known unsafe regions to allow full use of remaining options as suggested in the Cognitive Work Analysis approach (Vicente, 1999). The practical goal is not to keep operations outside known unsafe regions (constraints) allowing full use of the remaining options (as suggested by Vicente, 1999), but to keep operations as centrally located within known safe regions as feasible, while still carrying out a complex mix of goals. This is institutionalized through flight rules, which encode what is known about safe practices; by a conservative decision making culture; and by processes in which less-understood options can be explored and used in exceptional circumstances.

As a result, ADCO planning work is buffered from contact with physical constraints to a high degree. Known safe regions are used. For example, ADCO planners check that the specified attitude is within the region of attitudes validated by engineering analysis ("TEA approved") and certain operations are separated in time by standard amounts known to be sufficient. Because decision rules are known to be conservative, they are negotiable. Safe operation slightly outside of flight rules is possible and may be negotiated if it substantially furthers a mission goal and if assessment of the particular case suggests it is low risk. To further complicate matters, the (partly unknown) constraints of the ISS change fairly frequently, with payloads, vehicle docking, or new component deployment, requiring updating of the current-best-model of safe zones.

Thus, ADCO planners are not, and arguably should not be (given current knowledge), primarily driven by the physical constraints of the system whose control they are planning. In practice, neither construction of the plan nor executing the planned action is a matter of following a static procedure. Rather, thinking and judgment is required to negotiate a viable plan and to execute it, through many demands and tradeoffs. Therefore, finding one safe solution sufficient for the case at hand is paramount, and sticking closely to a known safe path is highly valued.

3) Social System: Known but flexible constraints.
A substantial part of the ADCO planning work concerns negotiating the developing plan with other groups. Many aspects of this concern coordinating timelines, from accommodating the 9 hour Houston-Moscow time difference to meeting the goals for the series of planning meetings as planned events draw close. In addition, approval protocol and information formatting must be accommodated. Yet here, even more than with physical constraints, the constraints are flexible in the short term and mutable over the long term. Timing of when information is exchanged can be negotiated through both formal and informal channels. A conspicuous and thought-requiring part of the task concerns getting and distributing information in a manner than facilitates short and long term cooperation, not (just) meeting a timeline. Once procedures have been agreed on, the goal is certainly to act within them and there are negative consequences of violations. However, acceptable "violations" can be negotiated if needed, and over the longer term, procedures could be changed and perfectly acceptable alternatives used.

Social systems generate, not simply follow, constraints. The agreements about communication and negotiation are in part constituted from the activity. Thus, these agreements and conventions are subject to change and should be considered as part of the system that is open to redesign. In particular, they should not be treated as immutable, fixed constraints within which activities and tasks must fit.

4) Constructed product: Generative domains.
Even though ADCO planning activity is closely linked to control of the ISS, the proximal product is a creation—a plan-- not the monitoring and control of an existing system. Activities that build structure, whether a plan, architectural drawing, or a document, are differently constrained. Many activities are a mix of generating new structure and accommodating existing structures. (Though related, this is not the contrast between coherence and correspondence domains, Vincent, 1990.)

5) Accessibility of Constraints.
In our circumstances, constraints look very different from the amount of liquid a tank can contain, or the maximum speed of a ship in particular conditions. Given these many complexities of the status and nature of constraints, starting by attempting to identify constraints may be inefficient for anyone, and impossible for a domain expert. Rather, beginning with a high-level task analysis may allow externalization and reflection on what is known about the domain. This in turn may allow identifying what type of constraining influences on activity there are, and how knowable, negotiable, and mutable they are. Iteration among understanding the boundaries of related activity and function, the tasks themselves, and the constraining factors over the domain may provide both the most efficient and accurate method of carrying out needs analysis.

### Limitations of the Case Study and Future Tool & Method Development

Our assessment of the tools and methods is limited by the rapid development and revision of methods took place during the case study. Our focus was on trying to aid the analysis rather than keeping records of our activity. No systemic assessment of effectiveness—such as comparison to an alternative method—was attempted. More broadly, our methods do not focus on ensuring every constraint or required activity is identified. Our methods stressed efficiency, not completeness. While our methods may still provide improvement, assessing completeness remains to be explored.

In addition, we identified limitations, and directions for improvement to several tools that we have not yet implemented. 1) Rather than providing just the organizing framework in the matrix, additional services could be included. Links among subtasks could be generated automatically aiding navigation and comparison among matrices. The information sources and content are identified in task descriptions, and these could be presented in menus, as many sources will recur across tasks. 2) Tools could automatically generate much of the task decomposition diagram from the information provided in the MDM. The MDM and task decomposition graph could be linked, and updating and revisions propagated across both representations. 3) Tools could explicitly pull out the information needs from the task matrix, and also allow information-to-task checking as well as task-to-information generation.

The support we provided for identifying constraints was modest. We believe that many constraints were identified during the process of task analysis, so one additional step might be supporting annotations about key functions and constraints, to be reviewed later. Additionally, it might be possible to adapt and support aspects of the Abstraction Hierarchy (Rassmussen, Pejtersen, & Goodstein, 1994).

Methods for vetting content with an expert can be improved. For example rather than just including constraints or tasks we think are accurate, we might include candidates we believe to be incorrect. This would require experts to discriminate and provide a more reliable, scalable method for experts to vet candidate representations. Explicit vetting cycles with the domain experts are very important, even if the object vetted was generated by the human factors group; this provides an opportunity to reflect, particularly though new representations, on the consequences, details, and overall structure of what was generated. More broadly, by developing quasi-formal representation and re-representation, information developed in one phase can be passed to and refined in other design phases.

As well as evaluation of the product of analysis though experts' vetting, we would like to evaluate our needs analysis method itself. Comparative evaluation of design methods is very difficult, but we are looking for opportunities to compare our approach to "current methods" (a needs analysis already being undertaken by others), assessing coverage and effort.

## Lessons Learned and Intended Beneficiaries

There are three morals we draw about applicability of our approach. First, logically, when constraints are negotiable (or unknown) in the short term, and mutable over the longer term, they cannot provide as strict a guide nor will they be so clearly identified; they do not stand outside and prior to the activities that both respond to and modify the constraints. Hence, prioritizing their identification may not be so useful as suggested by CWA. Second, methodologically, our iterative approach that starts with high-level task analysis may be widely helpful. Even for domain experts, information about what people do may be much more accessible than information about constraints, particularly when the constraints have the complex properties identified in the ADCO domain. Many cases may share these properties. Starting with task analysis, keeping it at a high level closely linked to mission, and iterating between characterizing tasks and characterizing constraints may be efficient and effective. Third, supporting these methods with simple tools may improve the effectiveness as well as efficiency of our approach. This case allowed use and modification of several simple supporting tools.

Our analysis methods require modest time commitments from domain and human factors experts. For example, domain and human factors worked together roughly 30 hours, including expert observation and vetting interviews. Yet the methods identified detailed task structure and domain constraints, which experts were initially unable to describe. As the methods are extended and supported by better tools, cost should decrease and information captured should increase. These methods should be particularly useful in situations with mutable constraints and boundaries, and where new systems are being built or old ones redesigned with limited resources.

We expect the method to be used collaboratively, and expect that analysis will require participation by both domain experts and human-system interaction experts. Designers and developers would become increasingly involved as the tasks and constraints take shape and solution approaches can be suggested. Ultimately, users should have better systems, because task and domain structure can be accommodated, even in projects with scarce resources for human factors, usability, and interaction design.

## Conclusions

Domains with flexible, mutable constraints and boundaries may be common. Domains may have these properties for several reasons, including extensive collaboration and attendant negotiation, or the feasibility of restructuring policies, practices, or physical infrastructure in the redesign, to a degree initially unknown. In addition, situations where work routinely involves generating new structures, rather than operating fixed-structure components (as in monitoring and control) will often have these properties.

In these conditions, our approach to needs analysis may be particularly helpful. Our methods specify beginning with a high-level task analysis and iterating between task analysis and domain constraint analysis, as well as shifting perspectives among alternative representations and among individuals with varying expertise. Tools supporting these methods can increase effectiveness and efficiency.

## Acknowledgements

We thank the ADCO controllers Jennifer Kimball, Erin Reed, and Tarik Ward for their participation, and Tarik Ward for extensive analysis work.

## REFERENCES

1. Beyer, H., and Holtzblatt, K. Contextual Design: Defining Customer-Centered Systems. San Francisco, CA: Morgan Kaufmann (1998).
2. Butler, K.A, Jiajie Zhang, J., Esposito, C., Ali Bahrami, A. Ron Hebron, R., & David Kieras. Work-Centered Design: A Case Study of a Mixed-Initiative Scheduler. In *Proc. CHI 2007*, ACM Press (2007).
3. Butler & Zhang (2009) Design models for interactive problem-solving context & ontology, representation & routines. In *Proc. CHI 2009*, ACM Press pp. 4315-4320
4. Diaper, D. & Stanton, N.A. (2004) *The Handbook of Task Analysis for Human-Computer Interaction*. LErlbaum Associates, Mahway, NJ.
5. Dowell, J. (1998) Formulating the Cognitive Design Problem of Air Traffic Management. *International Journal of Human-Computer Studies*. 49(5), pp. 743-766.
6. Kieras, D. E. (1996)Task analysis and the design of functionality. In A. Tucker (Ed.) *The Computer Science and Engineering Handbook (2nd Ed).* Boca Raton, CRC Inc.
7. Leveson, N. (1995) Safeware: System Safety and Computers. Addison-Wesley, New York, NY, USA.
8. McCurdy, M. (2009) Planning Tools for Mars Surface Operations: Humn-Computer Interactions Lessons Learne. IEEE AC.
9. Norman, D. A. (1986) Cognitive Engineering. In: *User Centered System Design*. Norman, D. A., and Draper, S. W. (Eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, USA, p. 31-61.
10. Sherry, L. and Feary, M. (2004) Task design and verification testing for certification of avionics equipment. In proceedings of: 23rd Digital Avionics Systems Conference. Salt Lake City, UT, October 24-28.
11. Sherry, L., K. Fennell, M. Feary, P. Polson (2006) Human–Computer Interaction Analysis of Flight Management System Messages. Journal of Aircraft. Vol. 43, No. 5, September–October 2006.

12. Rasmussen, J., Pejtersen AM, & Goodstein, L.P. (1994) *Cognitive Systems Engineering*. Wiley & Sons, New York, NY, USA.

13. Vicente.K. J. Coherence- and correspondence-driven work domains: implications for systems design, *Behaviour & Information Technology*,9:6,493-502 (1990).

14. Vicente.K. (1999). *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-based Work*. LErlbaum Associates, Mahway, NJ.