

11<sup>TH</sup> ICCRTS  
COALITION COMMAND AND CONTROL IN THE NETWORKED ERA  
Paper I-152

Constraint Recognition, Modeling, and Visualization in Network-Based Command and Control

*Topics:* Cognitive Domain Issues, C2 Analysis, Network-Centric Metrics

Rogier Woltjer\*

Cognitive Systems Engineering Lab  
Department of Computer and  
Information Science  
Linköping Institute of Technology  
SE-581 83 Linköping, Sweden  
Phone: +46 13 28 2289  
E-mail: rogwo@ida.liu.se

(\* STUDENT, Point of Contact)

Kip Smith

Division of Industrial Ergonomics  
Department of Mechanical  
Engineering  
Linköping Institute of Technology  
SE-581 83 Linköping, Sweden  
Phone: +46 13 28 2764  
E-mail: kipsm@ikp.liu.se

Erik Hollnagel

Cognitive Systems Engineering Lab  
Department of Computer and  
Information Science  
Linköping Institute of Technology  
SE-581 83 Linköping, Sweden  
Phone: +46 13 28 4043  
E-mail: eriho@ida.liu.se

# Abstract

This paper describes a method for the recognition of constraints in network-based command and control, and illustrates its application in a command and control microworld. The method uses Hollnagel's functional resonance analysis to extract the essential variables that describe the behavior of a command and control team. It juxtaposes these variables in state space plots that explicitly represent constraints and defines regions associated with alternative opportunities for action. Examples show how state space plots of experimental data can aid in the description of behavior vis-à-vis constraints. We discuss how state space representations could be used to improve control in network-based command and control settings.

## 1 Introduction

Command and control is in a phase of change. Both civilian emergency managers and military commanders seek to implement network-based organizations. The drive for network-based command and control stems from the identification of coordination problems in complex environments that resist solution by traditional hierarchic command structures (Cebrowski & Garstka, 1998). For example, civilian agencies want to improve inter-agency coordination (W. Smith & Dowell, 2000) and the military wants to overcome problems with combat power and speed. Moreover, joint operations by a growing number of civilian agencies (e.g. rescue services, local and national governments, and private sector agencies) and military services are often required to resolve emergency or conflict situations. The international nature of these emergencies, disasters, and conflicts impels civilian and military services to cooperate across traditional organizational and cultural boundaries. The prospect of having to coordinate activities across agencies, nations, and cultures suggests that a whole new range of technical and social problems may be waiting to emerge that can only be addressed by a network-based command-and-control structure.

The concept of network-based command and control is envisioned to enable forces (civilian or military) to organize in a bottom-up fashion and to self-synchronize in order to meet a commander's intent (Brehmer & Sundin, 2000; Cebrowski & Garstka, 1998). This concept steps away from the traditional 'platform-centric' hierarchical command structure where power is inherently lost in top-down command-directed synchronization. Network-based command and control envisions a high-performance information grid that makes sensor information available to the entire network. Part of this vision is that filtering, aggregation, and presentation of data will be done by computer systems. Nodes in the network are foreseen to be able to communicate with any other node, so that informed and appropriate action can be taken locally. The vision relies heavily on emerging information technologies.

The unit of analysis for studies of network-based command and control is much larger than for studies of traditional hierarchical control. In traditional organizations, the chain of command is orderly. The network-based vision entails a much greater flexibility in the implementation of communication and action. Large parts of the network necessarily become the unit of analysis. This situation presents a challenge to studies of network-based command and control. Two of the challenges are (1) how to describe and model joint behavior, and (2) how to design information technology that supports network-based command and control.

In this article we address these challenges by outlining a method for analysis of joint behavior using constraints and opportunities for action represented in state spaces. The method is meant to inform the design of information technology to be used in network-based command-and-control centers. Our research goal is to develop and verify a methodology to identify constraints generated by the actions of distributed collaborative decision makers. By addressing this goal, we aim to provide insight in distributed collaborative command and control in dynamic network-based settings. Understanding how constraints shape action of joint cognitive systems contributes to the science of network-based command and control and to the design of systems that support command and control. In this paper

we restrict our analysis to the management of emergency situations in distributed collaborative command and control.

## **1.1 Command and control and decision making**

The design of any support system is necessarily based on the model that the designers have of the task that is to be supported. Sometimes this model is only a chimera in the designers' minds. It would be distinctly preferable if they were to analyze the task they mean to support in detail, so that their design of their tool could be based on an explicit and well-informed model. Moreover, their design would benefit if they were to envision how their design will change the task and whether it will serve the actual purpose of the joint system that it is designed for (Hollnagel & Woods, 2005). The modeling of command and control and decision making is therefore a critical precursor of the design of support systems.

Diverse research traditions including, but not limited to, cognitive systems engineering, and decision making address the modeling of command and control. Here we offer a selective review of several models linking control, command and control, and decision making.

### **1.1.1 Joint command and control systems**

We adhere to the paradigm of cognitive systems engineering (Hollnagel & Woods, 1983, 2005). Cognitive systems engineering addresses questions such as how to make use of the power of contemporary technology to facilitate human cognition, how to understand the interactions between humans and technologies, and how to aid design and evaluation of digital artifacts. Related to distributed cognition (Hollan *et al.*, 2000) and macrocognition (Klein *et al.*, 2003), CSE takes an ecological view regarding the importance of the context when addressing cognition. It focuses on constraints that are external to the cognitive system (present in the environment) rather than on the internal constraints (memory limitations, etc.) that are foci of the information processing perspective. For CSE, it is these external constraints that determine behavior.

The unit of analysis in cognitive systems engineering is the 'joint cognitive system.' A cognitive system (Hollnagel & Woods, 2005) is a system that can control its goal-directed behavior on the basis of experience. The term 'joint cognitive system' (Hollnagel & Woods, 2005) means that control is accomplished by an ensemble of cognitive systems and (physical and social) artifacts that exhibit goal-directed behavior. In the areas of interest to cognitive systems engineering, typically one or several persons (controllers) and one or several support systems constitute a joint cognitive system which is typically engaged in some sort of process control in a complex environment.

An early account of modern command and control centers is provided by Adelson (1961), who describes decisions in what he called 'command control centers'. He states that "command centers are typically nodes in networks constituting command control systems" (Adelson, 1961, p. 726), where the word 'center' means 'a place where decisions are made', and does not imply centralization of decision making. This suggests that the network-based environments envisioned today are not as conceptually revolutionary as one may think. Adelson emphasizes that both the complexity and urgency of command in 1961 had increased compared to earlier times. Two outcomes of this were a need for a higher rate of information handling and more severe consequences of decisions. Adelson goes on to enumerate a large number of important factors that the "decider" must know about including the recent history of relevant variables' behavior, the present state of the world, predicted future states, including the opponents' behavior, the own system's and environment characteristics, alternatives for action, predicted outcomes, and objectives. Adelson (1961) even mentions war games and business games as tools to simulate and predict processes. Furthermore, he demonstrates problem-driven joint-systems thinking in that he emphasizes the importance of the objectives of technological improvement of command centers based on the real needs of "the assemblage of men and machines that constitute the system" (Adelson, 1961, p. 729).

## 1.1.2 The dynamics of decision making and control

Researchers often associate command and control with decision making. The term ‘decision making’ refers to a mental process that precedes the execution of action. The classical prescriptive model of the decision making process is expected utility theory. According to this theory, decision making includes four sequential steps: the generation of all possible options for action, independent assessments of the probability and utility of each option, and the selection of the option with the highest expected utility (see e.g. Von Neumann & Morgenstern, 1953).

Although expected utility theory has proven to be difficult to apply to dynamic, complex environments like command and control (Brehmer, 1992; Hollnagel, 1986, 1993, 1999; Klein & Calderwood, 1991), it has dominated the ‘expert system’ paradigm for decision support systems. Like expected utility theory generally, decision support systems based on the option generation and evaluation paradigm can perform well only in limited, well-defined and predictable domains. Predictably, they are typically inappropriate for network-based command and control, because the underlying model of ‘decision making’ does not fit the distributed nature of the task. The literature is replete with reports about the pervasively negative consequences of support systems based on decision theoretic designs including, but not limited to, lack of user acceptance, brittle performance when faced with unanticipated novelty, users’ over-reliance on the machine’s ‘expertise,’ and biasing users’ cognitive and decision processes (Bainbridge, 1983; Dekker & Woods, 1999, 2002; Hollnagel, 1999; Hollnagel & Woods, 2005; Roth *et al.*, 1997).

Fortunately, alternative models have emerged that are more appropriate for command and control applications. Numerous studies of decision makers in so-called ‘naturalistic’, high-stakes, time-critical, and complex settings have led Gary Klein and his colleagues to propose a model they call ‘recognition-primed decision making’ (e.g., Klein & Calderwood, 1991). Their critical insight is that ‘decision makers’ in naturalistic settings actually don’t make decisions and only rarely consider more than one option. The recognition-primed decision making model states that decision makers generate one, and only one, option based on recognition of familiarity, evaluate whether that option will work (using ‘mental simulation’), and implement if it will. If it will not work, they generate another option and evaluate it. The process cycles until a satisfactory option is found. Adequate recognition and simulation of the situation are thus the keys to successful ‘decision making’ according to this model.

A second approach is called ‘dynamic decision making’ (Brehmer, 1992). This approach focuses on the functions served by decision making in order to gain control, or to achieve some desired state of affairs, rather than on the decisions themselves. As its name implies, dynamic decision making involves tasks with a dynamic character. This implies that a series of interdependent decisions in real-time is required to reach the goal, and that the state of the decision problem changes both autonomously and as a consequence of the decision maker’s actions. This approach describes decision making as the intersection of two control processes: (1) the system that the decision maker aims to control and (2) the means for achieving control. The second process is used to control the first (Brehmer & Allard, 1991). The DOODA loop (Brehmer, 2005) is a contextualized model of this control loop approach, integrating a loop of (1) sensemaking, decision making, action, and information collection, with (2) more military-specific concepts such as command, order, mission, and sensors into a comprehensive model of command and control.

The third approach to modeling that we review are the models that form the foundations of cognitive systems engineering (CSE, Hollnagel & Woods, 2005): (1) the cyclic model of action as it is performed by joint cognitive systems, (2) a contextual control model (COCOM), and (3) the extended control model (ECOM) of cognitive systems engineering. The cyclical model of human action is the heart of cognitive systems engineering, and is based on Neisser’s (1976) perceptual cycle. This model, also called CAEC loop, basically states that an operator’s current understanding (or construct, C) directs the operator’s next action. This action (A) produces some information or feedback. This information (or event, E) modifies the operator’s current understanding, etc. This does not mean that these concepts always occur in this order: Event evaluation and action often occur in an intermingled or iterative

fashion. The control principle is based on the idea that behavior towards goals, and thereby control, is a combination of feedback (compensatory) and feedforward (anticipatory) control. The four simultaneous control loops in ECOM range in their character on a gradual scale from short-term compensatory control to long-term anticipatory control.

## **1.2 Control support through the recognition of constraints**

These reflections result in the cognitive systems engineering view that states: “From the joint cognitive system perspective, the intelligence in decision support lies not in the machine itself (the tool), but in the tool builder and in the tool user.” (Woods, 1986, p. 173). The cognitive systems engineering view on intelligent decision support for process control is highly influenced by the principle that a joint cognitive system controls its behavior to achieve its goals. Any analysis of the performance of joint cognitive systems should rely on directly observable phenomena. The unit of analysis is therefore not decision but action. Rather than supporting decision making, cognitive systems engineering aims to support actions directed at gaining and maintaining control. In this view interaction between the various parts of the joint cognitive system has to be designed to enable the joint cognitive system to cope with complexity and to handle disturbances on the way to retaining control. This view recognizes that process control environments are often fundamentally complex and that oversimplifying the interaction would likely reduce the joint cognitive system’s ability to maintain control. Supporting control means supporting the evaluation of the situation and anticipation of future events, the selection of action, and the performance of action, as well as ensuring that the time available for actions is sufficient (Hollnagel & Woods, 2005).

In the many tasks where the option generation and evaluation paradigm of intelligent decision support systems seems inappropriate, other ways of supporting the joint cognitive system to gain and maintain control may be found. For the domain of air traffic control, Dekker & Woods (1999) mention an alternative form of supporting controllers: “In one situation, controllers suggested that telling aircraft in general where *not* to go was an easier (and sufficient) intervention than telling each individual where to go” (p. 94). This statement suggests that operators may necessarily and sufficiently be informed about the constraints on their actions, rather than all possible actions. Fundamental similarities between the air traffic control and command and control tasks indicate that this idea may transfer well to the command and control domain.

Similarly, the approach to decision support that we are currently investigating is to support control through recognition of constraints. The theoretical basis for this has been outlined in the sections on the rationale for looking at constraints to analyze behavior, and to support goal-directed behavior. Particular models of command and control however also encourage the investigation of a constraint-based approach to the analysis and support of joint systems behavior in command and control.

Dynamic decision making research in Brehmer’s view “is concerned with people’s formulation of goals and models as a function of the observability and action possibilities of the system to be controlled” (Brehmer, 1992, p. 218). Thus the question what constitutes observability and action possibilities is a research problem. The discussion of constraints above is exactly concerned with this problem. Constraints influence the action possibilities. Hence, it is important that people’s model of the system to be controlled is adapted to the possibilities for action and therefore the constraints on action to be able to control. To be able to know the possibilities for action, they need to be observable. As mentioned above, exhaustive option generation is not meaningful in many complex environments with conflicting goals and multiple ways to achieve these goals. Therefore, to be able to adapt one’s model to the possibilities for action, it may be more useful to be able to clearly and more directly observe the constraints on action. This is the perspective on command and control support investigated in this paper, in the form of visualization of constraints.

The constraint concept has recently been discussed in a command and control context. As a result of a study employing grounded theory, Persson (1997) defines command and control as “the enduring management of internal and external constraints by actors in an organization in order to achieve

imposed and internal goals” (p. 131). He argues that the core of command and control is “constraint management”, and defines constraint as something that is an obstacle to action or goal-achievement. Persson indicates the importance of the recognition, resolution, and thereby management of constraints in command and control.

### 1.3 Constraints and their representation

The concept ‘constraint’ has been addressed by scientists from a variety of disciplines and perspectives and has implications for cognitive systems engineering (Ashby, 1956; Checkland, 1981; Gibson, 1986; Hollnagel & Woods, 2005; Leveson, 2004; Norman, 1998; Reitman, 1964; Vicente, 1999; Vicente & Rasmussen, 1992; Woods, 1995). Combining these perspectives we define **constraint** as either a limit on goal-directed behavior, or an opportunity for goal-directed behavior, or both. Constraints are invariably described as essential factors in the functioning of (cognitive) systems. This work addresses constraints explicitly in the analysis of behavior of joint cognitive systems. In CSE constraints are said to shape the selection of appropriate action (Hollnagel & Woods, 2005). Similarly, the related disciplines of cybernetics and systems theory consider the concept of constraint to be of major importance, because constraints facilitate control.

In cybernetics, constraints are recognized to play a significant role in process control. One of the fundamental principles of control and regulation in cybernetics is the Law of Requisite Variety (Ashby, 1956). This law states that a controller of a process needs to have at least as much variety (behavioral diversity) as the controlled process. Constraints in cybernetics are described as limits on variety. Constraints can occur on both the variety of the process to be controlled and the variety of the controller. This means that if a specific constraint limits the variety of a process, less variety is required of the controller. If a specific constraint limits the variety of the controller, less variety of the process can be met when trying to exercise control. Regarding the prediction of future behavior, Ashby notes that if a system is predictable, then there are constraints that the system adheres to. Knowing about constraints on variety makes it possible to anticipate future behavior of the controlled process, and thereby facilitates control.

The recognition of constraints on action has been observed in field studies as a strategy to cope with complexity in socio-technical joint cognitive systems. Example domains include refinery process control (De Jong & Köster, 1974), ship navigation (Hutchins, 1991a, 1991b), air traffic control (Chapman *et al.*, 2001; K. Smith, 2001), and trading in the spot currency markets (K. Smith, 1997). Similarities of system complexity and coupling in these domains and command and control suggest that the recognition of constraints is likely a useful approach in command and control as well.

Woods (1986) emphasizes the importance of spatial representations when providing support to controllers. In this paper we develop the visualization of constraints as the support strategy. Representation design (Woods, 1995) and ecological interface design (Vicente & Rasmussen, 1992), offer design guidelines concerned with constraint: Decision support systems should facilitate the discovery of constraints, represent constraints in a way that makes the possibilities for action and resolution evident, and highlight the time-dependency of constraints. One representation scheme for such discovery is the state space.

State space representation is a graphic method for representing the change in state of process variables over time (Ashby, 1960; Flach *et al.*, 2003; Knecht & Smith, 2001; Port & Van Gelder, 1995; Stappers & Flach, 2004). The variables are represented by the axes. States are defined by points in the space. Time is represented implicitly by the traces of states through the space. Figure 1 is an example state space in which the variables are fuel range and distance. The regions with different colors represent alternative opportunities for action. The lines between these regions are constraints on those actions. The arrows represent traces of states of a pair of processes through the space as their states change over time. The opportunities for action have yet to change for the first process, but they have changed for the second.

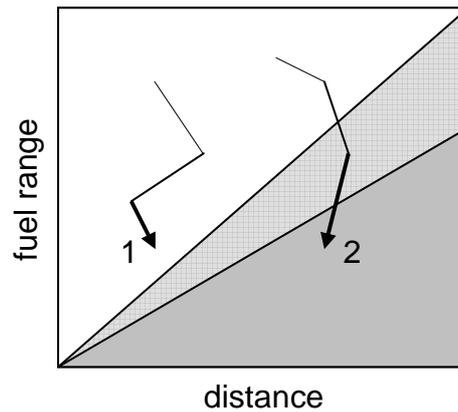


Figure 1 A generic state space. Constraints are represented as thresholds between regions associated with alternative opportunities for action. The arrows represent a pair of processes as their states change over time.

Recent work in cognitive systems engineering, human factors, and ergonomics reflects a renewed interest in the state space representation. Stappers & Flach (2004) describe state space representation as a promising and rich visualization method of the behavior of cognitive systems. They state that it steps away from the typical block diagrams of perception, mental processing, and motor control that hinder appropriate thinking about dynamics. They describe the metaphor of state spaces as “that of the cartographer’s map, showing a landscape with roads and pitfalls, mountains and rivers, i.e., opportunities and threats, but leaving the reader’s mind free to roam and imagine developing states and possible routes through the territory” (Stappers & Flach, 2004) (p. 825). In sum, state spaces ignore processing stages and activities and emphasize constraints.

## 1.4 A method for the recognition of constraints

From the discussion above, we can conclude that constraints are acknowledged in work practice and described in the literature to shape action and enable control, and that being aware of constraints is essential for process controllers. However, in complex and dynamic environments where constraints continually change, where people have to work collaboratively and where information and people are distributed, keeping abreast of all relevant constraints can be difficult. To address this difficulty, this paper investigates a method for recognizing constraints in process control tasks. The method is intended to become part of the support systems that Brehmer & Sundin (2000) and Cebrowski & Garstka (1998) envision.

The method has three steps, which are discussed in turn.

(1) The first step in the method is a functional resonance analysis of the joint cognitive system’s process control task. The analysis is conducted by (or in consultation with) a domain expert.

The modeling of the joint cognitive system’s process control task is done by connecting modules that represent the functions that need to be performed. All functions are described through and may be linked via their aspects, in terms of their input and output, preconditions, resources, time, and control. This functional modeling scheme was recently developed by Hollnagel (2004) for accident modeling and complex system analysis purposes under the acronym FRAM (functional resonance accident model). The modules of functions and their characteristics are therefore called FRAM-modules. Because the modeling method is developed for complex joint human-machine system analysis, it is also suitable for modeling the tasks and functions that are performed in network-based command and control, which typically involves such complex systems.

Figure 2 describes the six aspects that a FRAM-module addresses for each function that is identified. To find the FRAM-modules, one may start with the top-level goal, which may translate into the top-level function, or one may start with any function and move on to related functions. Typically, the goal

in this domain is to get some sort of process under control, such as a progressing fire, a crowd of panicking people, a spreading oil spill, or a moving adversary military force. This translates into the top-level function of for example fighting the fire, directing the flow of people, blocking and cleaning the oil spill, or fighting the adversary forces. From there on, the functions that are necessary or otherwise related to the input of the function, its timing, control, required resources, and preconditions, determine the other functions in the model of a task. The tasks identified through a goals-means task analysis (GMTA; Hollnagel, 1993) may aid in the definition of functions in FRAM.

<i>Function Aspect</i>	<i>Description</i>
Input	That which the function uses or transforms
Output	That which the function produces
Preconditions	Conditions that must be fulfilled before the function can be carried out
Resources	That which the function needs or consumes when it is carried out (e.g. matter, energy, information, manpower)
Time	Time available, as a resource or a constraint
Control	That which supervises or adjusts the function (e.g. controller, guideline, plan, procedure)

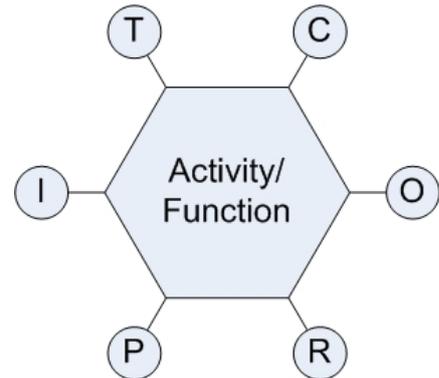


Figure 2 FRAM-module description of function aspects and hexagonal function representation (Hollnagel, 2004).

FRAM has possibilities and advantages similar to goals-means or means-ends task analysis, which is a highly related method. Both FRAM and goals-means task analysis (GMTA) have a recursive structure, focus on the goals of the joint human-machine system rather than strict function allocation, and can be applied to operational as well as hypothetical systems. The Test-Operate-Test-Exit unit (Miller *et al.*, 1960) and the means-ends analysis in the General Problem Solver (Newell & Simon, 1961) are early examples of goals-means task analysis. More recently, the principles of GMTA have successfully been applied to the modeling, design, and evaluation of process control tasks (Hollnagel, 1993; Hollnagel & Bye, 2000; Hollnagel & Woods, 2005; Lind & Larsen, 1995).

(2) The second step in the method is to identify the essential variables, the subset of process variables that the joint cognitive system can both observe and (indirectly) affect.

This step in the method identifies the variables that change in the performance of the tasks identified in the FRAM analysis. These are the variables that are affected by the joint cognitive system's goal-directed actions, and the variables upon which the joint cognitive system bases its understanding of the situation. These variables stem directly from the components of the FRAM module, that is, the constraints associated with time, control, preconditions, and resources can be identified by addressing these aspects of each function.

Following Neisser (1976) and Hollnagel (2003), support for people doing their work should focus on assuring a satisfactory understanding of the situation and an effective execution of action toward the goals of the joint cognitive system. The act of controlling steers the process to be controlled toward the fulfillment of the joint cognitive system's goals. The act of controlling is modeled as an assembly of feedback (compensatory) and feedforward (anticipatory) loops (Hollnagel, 2003). Feedback and feedforward control, and the understanding of the situation, are based on observations of the state that the system is in.

In the process control domains that are of interest to cognitive systems engineering, the units of observation are typically states, defined by the current instantiation of a large number of variables that can change rapidly over time. Following Ashby (1956), who used the term ‘essential variables’ for the variables that must be kept within assigned limits for an organism to survive, we use the term ‘essential variables’ for the subset of process variables that the joint cognitive system can both observe and (indirectly) affect. We imply that essential variables must be known for the joint cognitive system to function adequately.

(3) The third step of the method has three parts: (a) sampling the values of the essential variables (b) juxtaposing these variables to form a set of multidimensional state spaces, and (c) comparing the location of the data in the state spaces to the thresholds between regions specifying differing opportunities for action.

Step 1, the FRAM analysis, identifies the functions for and constraints on the execution of a task, and step 2 identifies the variables that change during its execution and the constraints on these variables. Step 3 juxtaposes variables associated with related tasks to form state spaces that make these preconditions explicit by specifying regions where the constraints are (not) met. Boundaries between regions form thresholds that represent constraints on action. In the example of Figure 1, there are two such thresholds that divide the space into three regions.

In the remainder of this article we discuss an experiment that provides an existence proof that the representation of data and constraints in state spaces can form the basis for the design of information systems to support operators in command-and-control environments.

## 2 Application of the constraint recognition method

This section describes the microworld study that we have conducted, and more importantly the insights we have gained regarding the application of our method for the recognition of constraints.

### 2.1 *Microworlds and C3Fire*

Microworlds are simulated task environments that (a) provide a task that can be made more complex, challenging, and realistic than traditional laboratory studies but that (b) generalize to interesting parts of real world problem solving while remaining (c) are more controllable, tractable, reproducible, and flexibly designable research environment than a field study (Brehmer & Dörner, 1993; Funke, 2001; Gray, 2002).

C3Fire (Granlund, 2002) is a fire-fighting microworld in which a group of people work together to gain control of a computer-simulated fire. Their task is to collaborate in an experimentally controlled configuration for command and control interaction, under observation of a researcher who manages the experiment. Figure 1 presents an overview of the C3Fire microworld. Its elements include a map showing vegetation, buildings, vehicles, and the fire, and the computer network for controlling the vehicles and communicating by e-mail.

In the C3Fire setting used in this experiment, various mobile units and stationary units are located on the map. Mobile units (trucks) need fuel to move across the map. Stationary units have a fixed position on the map and cannot move. Fire trucks are mobile units that can fight fires. To do so, they need water. Water providing units can provide other units with water. This class is formed by water trucks and water stations. Similarly, fuel providing units are units that can supply all moving units with fuel. These can be divided into fuel trucks and fuel stations. The class of stationary units consists, besides of water and fuel stations, of vegetation and buildings.

The rate of burning and spreading of the fire can be set by the researcher, and is typically made dependent on vegetation, terrain, the presence of buildings, and wind direction and speed. Thus, these properties of the environment constrain the development of the fire.

Participants direct the units where to drive by interacting with an interface showing a map with the dynamic simulation, an e-mail tool, and some data describing the state and characteristics of the trucks.

As the discussion above has illustrated, the C3Fire microworld captures many of the characteristics of complex environments: It implements high degrees of connectivity, complexity, uncertainty, time pressure, and polytely. It provides a test bed for research on collaborative and distributed command and control in a dynamic, volatile, uncertain environment. By allowing several participants to control the interdependent trucks, their actions become mutually enabling and constraining. To sketch the method for the recognition of constraints and to illustrate behavior in terms of constraints, this paper discusses the application of the method to an experimental study in C3Fire, and deals with the issue of how support systems' designers could take advantage of the representations it produces. Publication of more detailed analysis and results of the study is in preparation and can be found elsewhere (Woltjer, 2005; Woltjer & Smith, 2005).

The various units in C3Fire are mutually enabling and constraining because of interdependencies in the consumption and provision of water and fuel. Interdependencies among decision makers arise whenever different classes of units are assigned to different participants in the simulated command and control center. For example, if fire trucks need water and fuel, water trucks and fuel trucks constrain the actions of fire trucks. If different people have control over these different units, their actions are mutually enabling and constraining. This also becomes apparent through the first step in the method described above, the goals-means task analysis.

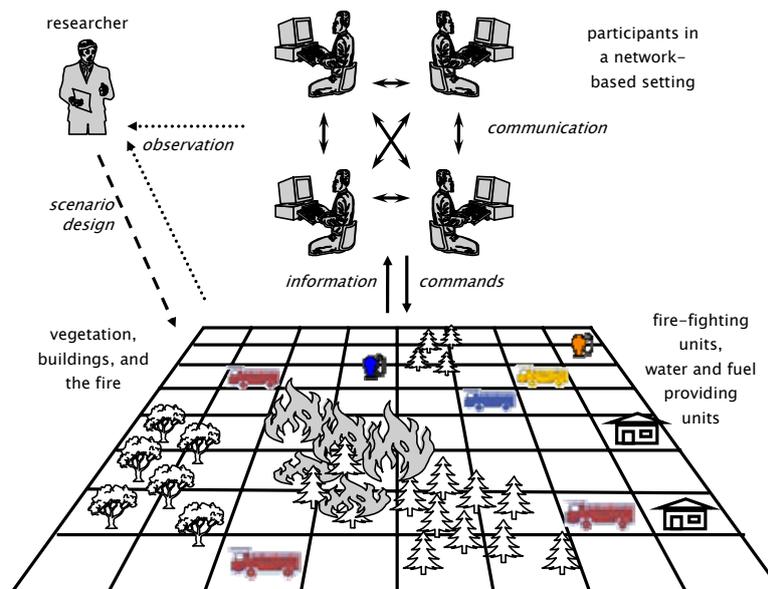


Figure 3 The conceptual C3Fire microworld setting used in the present study, adapted from (Granlund, 2002).

## 2.2 The constraint recognition method applied to C3Fire

This section presents the application of the constraint recognition method to the C3Fire microworld.

A small part of the functional modeling task analysis of C3Fire is shown in Figure 4. The analysis reveals how functions are linked and thereby how units are mutually enabling and constraining. To take the example of the 'fire trucks fight fire' function, the input of this function is burning vegetation and buildings, which may be saved from or lost to the fire (the output of the function). The preconditions of fighting a fire are (1) that the fire truck has water (which is the output of the water refilling function) and (2) that the fire truck is collocated with that fire (which is the output of the moving-to-fire function). The time and pace of fire trucks fighting fire is influenced by the fire spread rate and fire fighting rate for example. The main resource used for fire fighting is water.

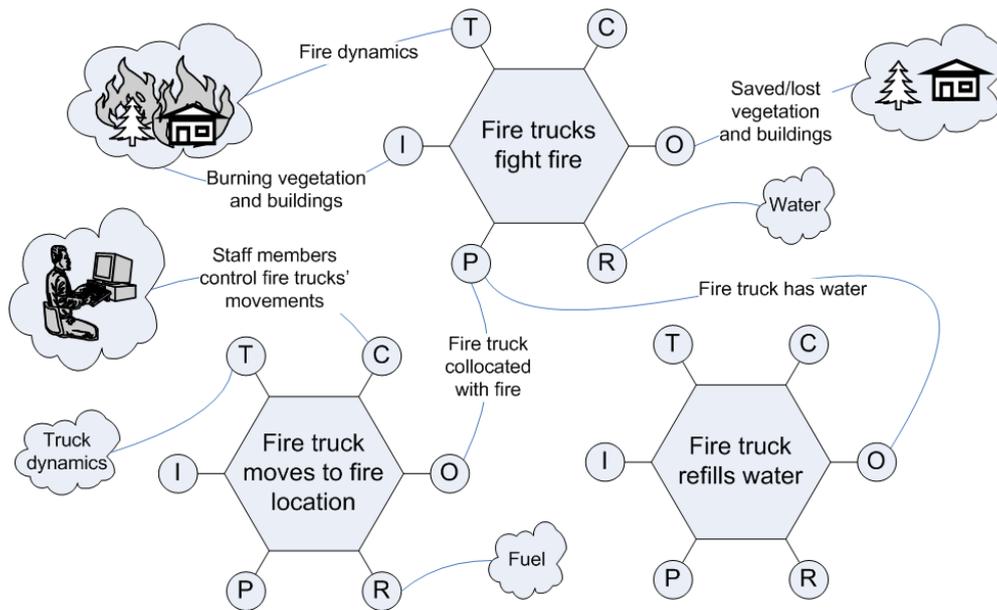


Figure 4 Functional modeling (FRAM) analysis of a small part of the task in the C3Fire microworld.

The essential variables were subsequently analyzed according to the second step in the method. These are variables that are both affected by the actions of the network of participants, and used by the participants to assess the situation that unfolds during the experimental scenarios. They include fuel and water levels, distances to intended positions and other trucks and stations, time available to fight fire and move across the map, etc.

Table 1 presents an example of an analysis of essential variables linked to the various aspects of a FRAM-module. It describes the function of a truck  $T$  refilling its resource  $R$  from a source  $S$  as an example. This abstract description of the function may then be instantiated with for example ‘fire truck F4’ for truck  $T$ , ‘water’ for resource  $R$ , and ‘water truck W12’ for source  $S$ . Table 1 also illustrates how operators such as ‘increase’, ‘level’, ‘same’, ‘time’, ‘between’, and ‘rate’, may be used to express actions on or properties of the various quantities in the simulation. These operators enable the linking of aspects of one function to the other function. For example, changing the position of truck  $T$  or source  $S$  in order to fulfill the precondition ‘same(pos(truck  $T$ ), pos(source  $S$ ))’ is the output of the truck moving function, which in turn specifies the indirectly related (and therefore marked with an asterisk) variable of ‘\*time(between(truck  $T$ , source  $S$ ))’.

<i>Truck <math>T</math> refills resource <math>R</math> from source <math>S</math></i>	<i>Description</i>	<i>Essential variables, examples</i>
Input	truck $T$ source $S$ resource $R$	coordinates of truck $T$ coordinates of source $S$
Output	increase(level(truck $T$ , resource $R$ ))	new level(truck $T$ , resource $R$ )
Preconditions	same(pos(truck $T$ ), pos(source $S$ )) level(source $S$ , resource $R$ ) > 0	pos(truck $T$ ), pos(source $S$ ) *time(between(truck $T$ , source $S$ )) level(source $S$ , resource $R$ )
Resources		
Time	refill dynamics	rate(increase(level(truck $T$ , resource $R$ , source $S$ )))
Control		

Table 1 Description of the function module ‘truck  $T$  refills resource  $R$  from source  $S$ ’ with examples of essential variables.

Experimentation with C3Fire comprised multiple trials where different scenarios of a simulated fire-fighting task were presented to a multi-person team. Thirty-two Swedish with a mean age of 24.6 years participated in the study for a monetary compensation. The participants were not trained in command and control. The participants filled out a form stating their informed consent to anonymous participation in the study. There were a total of 8 scenarios with changing geographical layout and task difficulty. The purpose of these manipulations was to keep the scenarios challenging and the participants engaged. The 32 participants engaged in the C3Fire scenarios in 8 four-person teams, each playing 8 scenarios independently so that 64 trials of empirical data were collected. C3Fire captures and logs a large number of variables. Performance variables include the number of squares that were still burning, closed out, or burnt out at the end of each scenario, which will not be discussed further here. For the purpose of the analysis of constraints, all variables associated with the fire truck, water truck, and fuel trucks' behavior were logged.

We will now turn to the description of the application of the third step of the method for recognizing constraints. This step entails the sampling of the values of the essential variables, juxtaposing these variables in multidimensional state spaces, and comparing the location of the data in the state spaces to the thresholds between regions specifying differing opportunities for action. Our discussion emphasizes the state space representation of behavior relative to constraints, tells the stories the state spaces contain, and illustrates the utility of state space representations in a command and control environment.

An example of a state space for constraints is shown in Figure 5. This figure juxtaposes the variables 'distance to the nearest fuel providing unit' and 'fuel range' and plots data of all fire trucks (F1-6) and all water trucks (W7-9) during one entire trial. The distance to the nearest fuel providing unit is the distance of a unit to the nearest fuel truck or fuel station. The fuel range indicates how many squares the truck can drive with the current fuel level. The higher black line indicates the states where the truck's fuel is exactly sufficient to reach the nearest fuel providing unit. In the region above this line, the distance to the nearest fuel providing unit is smaller than the distance that the current fuel level allows the truck to drive. In other words, when a truck is above the line, it can make it on its own to a fuel providing unit. In the region below the lower constraint line, the truck does not have sufficient fuel to reach the nearest fuel providing unit, and a fuel truck will have to come to it instead to refuel. It may be a strategy at a particular point in time that the fire trucks in a specific area with a small fire have to refuel themselves because fuel trucks are occupied elsewhere, and that fuel trucks in another area have to actively supply fire trucks with fuel because they are too occupied fighting extensive fires. This state space representation thus links the goal structure of the task, the constraints of the environment, strategies for performing the task, and the actual states of the control process in relation to the three former concepts.

Between these constraint lines, there is a grey zone where it is not certain whether the truck can reach its intended position. The existence of the grey zone in the state space reflects the uncertainty that is often encountered in real world traveling, where one often does not know the exact fuel consumption. Often however one can estimate the upper and lower limits on fuel use. The exact fuel consumption is difficult to predict exactly because of local influences such as road condition, weather, etc. In this way C3Fire captures some of the inherent uncertainty in the command and control domain.

Strictly horizontal lines mean that the truck is not moving (fuel range stays the same) and the nearest fuel providing unit is moving either towards or away from the truck. In cases when the fuel providing unit moves away, another fuel providing unit may become the nearest.

The vertical line in these state spaces at  $X = 1$  reflects the fact that when trucks refuel, they are one square removed from the nearest fuel providing unit. As they refill their fuel range increases to a maximum of 20, tracing the vertical line at  $X = 1$ .

In Figure 5, we can again see that trucks generally had sufficient fuel ranges to make it to the nearest fuel providing unit. Examination of the trace of truck W7 through the state space is informative. The

oval in Figure 5 shows W7 at states (6.5, 8) and (6.5, 7), where it had enough fuel to reach the nearest fuel providing unit. Between these states, W7 kept moving (decreasing fuel level) but the line goes straight down, indicating that the distance to the nearest fuel providing unit stayed constant. It can be inferred that the nearest fuel providing unit was moving in the same direction as W7. Thereafter the distance to the fuel providing unit grew, and truck W7 then made a long excursion into the region of the state space that indicates that it could not reach a fuel providing unit. It crossed both of the black constraint lines, eventually reaching a state with approximate coordinates (7.2, 3), highlighted by the circle in Figure 5. At this time, the horizontal line indicates that a fuel providing unit approached truck W7. Thereafter, the diagonal line from (5.8, 3) towards the origin indicates that W7 started to burn fuel as it moved towards the fuel providing unit.

The vertical lines at the bottom of the figure that intersect the horizontal axis reveal that trucks F2, F4, F5, F6, W7, and W8 all ran out of fuel one or more times. They were not positioned next to a fuel providing unit while their fuel ranges were zero. In these cases a fuel truck would have to come to them. Knowing this information in advance or being able to predict when this situation may happen would have been a huge advantage to participants trying to engage in anticipatory constraint control (De Jong & Köster, 1974).

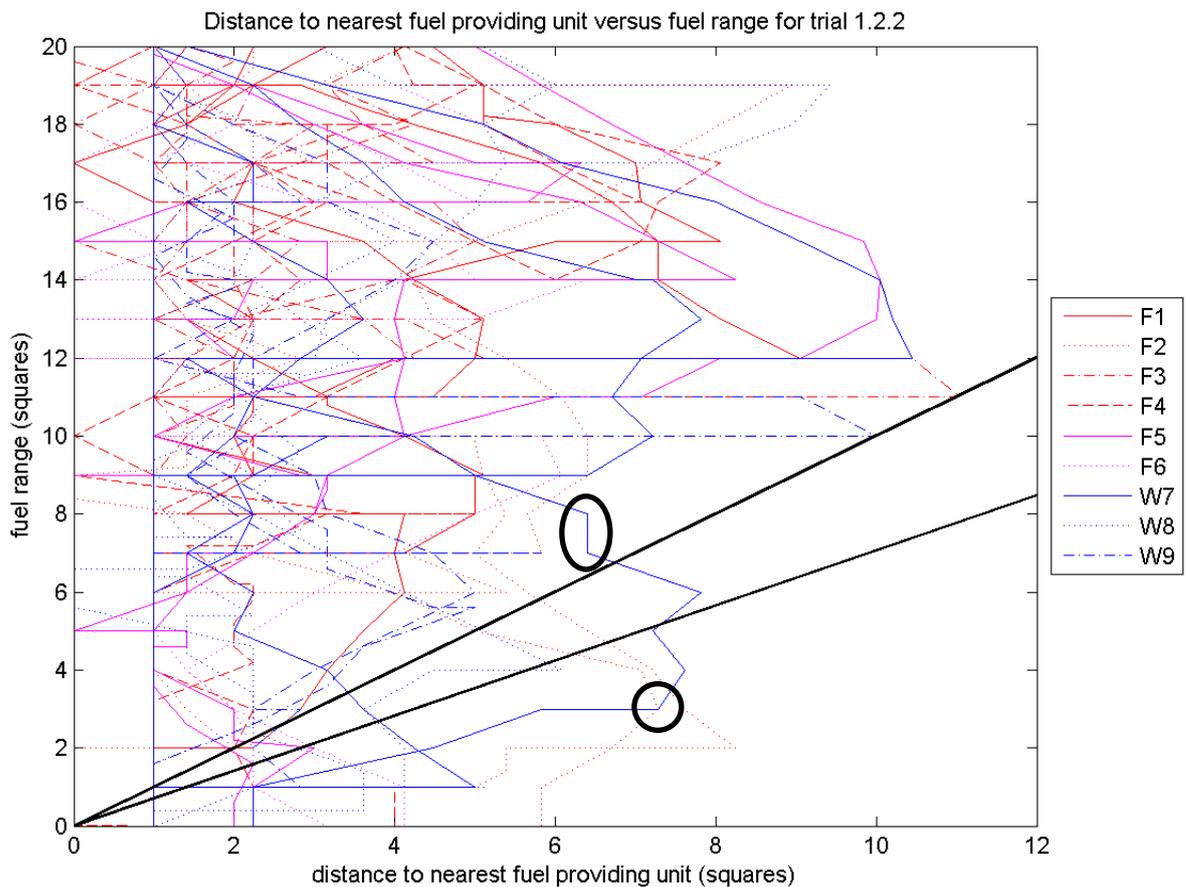


Figure 5 Sample state space for the variables ‘distance to the nearest fuel providing unit’ and ‘fuel range’ for fire trucks and water trucks.

The second example state space we discuss here is the space that is formed when ‘transit time for water resupply’ is juxtaposed with ‘fire fighting time given the water supply’ (current water level). Such a space is shown in Figure 6. The transit time for water resupply is defined as the number of seconds it takes to bridge the distance between a fire truck and the nearest water providing unit (either a water

truck or a water station). The fire fighting time given the water supply is defined as the number of seconds that fire can be fought until the water level is zero.

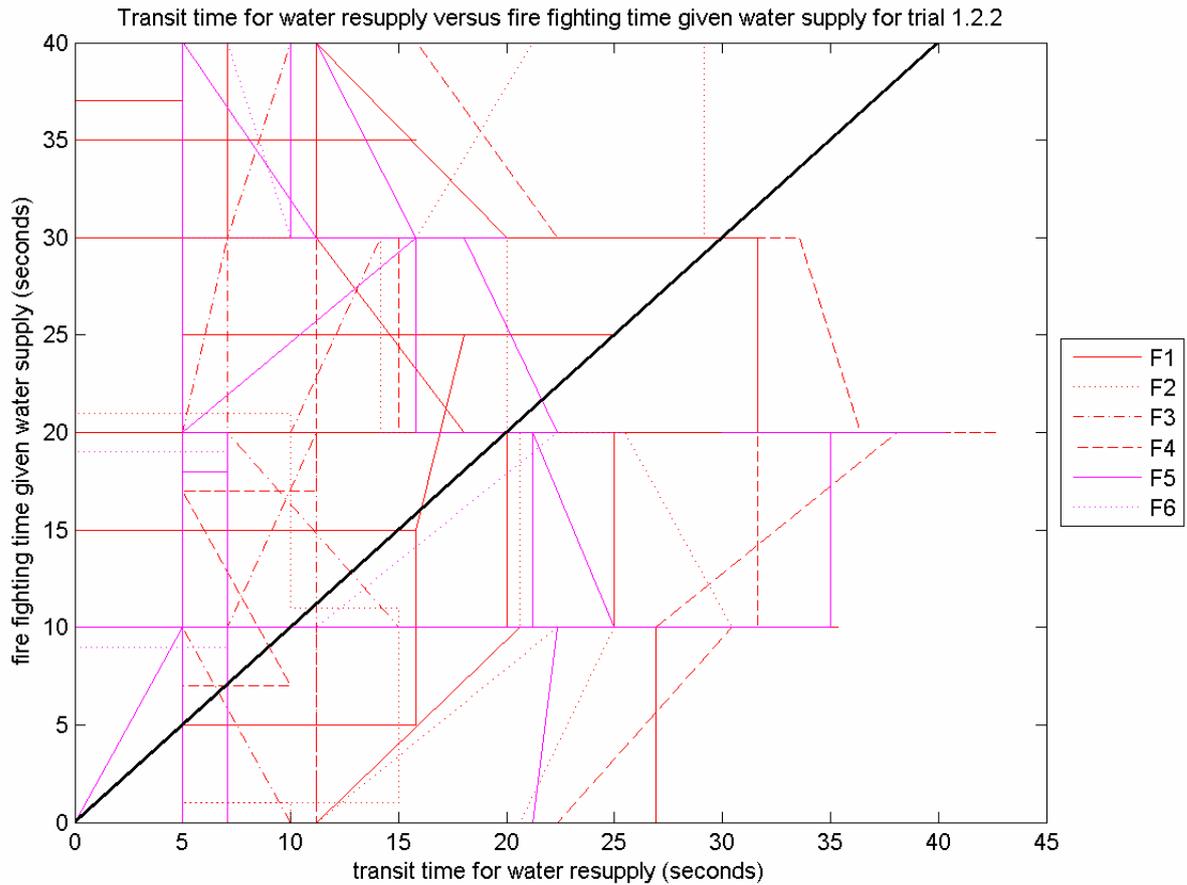


Figure 6 Sample state space for the variables ‘transit time for water resupply’ and ‘available fire-fighting time’ for fire trucks.

In discussing one possible use of this space we will sketch a strategy employing two assumptions about the desired functioning of fire and water trucks. First, since fire trucks are already quite engaged and busy fighting fire, it may be distracting to have to constantly think about their water level and where, when and how to get water. This may be especially unnecessary since the water trucks are dedicated to water provision. Therefore, it is likely a good strategy for water trucks to bring water to fire trucks. Second, fire trucks often try to fight fires in a series of adjacent cells, and normally several adjacent cells are on fire. This makes it useful to talk about the fire fighting time for which water is available even when it reaches quite high numbers.

These considerations make it reasonable to create a threshold defined by states where the available fire fighting time is equal to the time it takes for a water truck to reach a fire truck. This line is drawn in the state space of Figure 6. Above and left of this line, there is more fighting time available than it would take for a water truck to arrive. Well above or left of this line, there is little need to refill with water, since there is plenty of water left with respect to the distance to the water supply. Below and right of this line, the fire truck will run out of water (given continuous fire fighting) before it can be refilled. Assuming that the fire truck will have to continue to fight fires in nearby cells, action probably has to be taken to get water providing units closer to the fire truck to enable it to continue to fight fire.

In the trial shown in Figure 6 the transit time for water resupply often far exceeded the fire fighting time available. This applies to almost all fire trucks. Moreover, the lines descending vertically into the

horizontal axis illustrate that trucks F1, F4, F5, and F6 were all without water at times when it would have taken considerable time for a water truck to get to them. It is not possible to evaluate the quality of this team's performance from the state space of Figure 6 alone. Letting fire trucks wander beyond their water supply may have been part of a strategy, or it may not have mattered in the specific situation. The information made available by the state space would have likely been useful to the participants in case these situations were unanticipated or unintended.

In the discussions of state spaces, fuel ranges have been a central part of the analysis. Since maintaining fuel range at appropriate levels seems so critical, we illustrate that there are other visualization methods that may be useful to explore in the analysis of constraints. We have created snapshots that show the fuel range values plotted as circles around the positions of the trucks. Figures 7 and 8 present two examples of the utility of these snapshots. The two figures show data from different times during the same trial. They relate essential variables (position, fuel range), and make changes in their values over time salient. What emerges from these plots is the presence or absence of overlap between fuel ranges. Overlaps in fuel ranges mean that trucks can reach each other, which is desirable given that fuel is a critical resource. In contrast, effectiveness and efficiency are likely to be hampered when there is little overlap. In Figure 7 there is considerable overlap; most trucks can get fuel. In Figure 8, there is little; the situation is hopeless.

In Figure 7, after one minute of running the scenario, the fuel ranges of most of the trucks overlap. They can reach each other to refill water and fuel. This representation of fuel constraints makes the dire situation of truck W9 jump directly into attention. We can recognize instantly that the truck was quite remote and had a fuel range of zero (as illustrated by the small circle around the position of the truck). It had no means to get back to the cluster of trucks grouped together where the action is. This figure makes it easy to recognize what to do: someone must supply truck W9 with fuel, so that it can get back into the game.

Figure 8 plots the fuel ranges 15 minutes later ( $t=960$ ). The situation is more or less hopeless. Only truck F3 can still move. The others are scattered across the map and have fuel ranges of zero. Nothing can be done to remedy this situation since all fuel trucks have run out of fuel.

### 3 Discussion

A large pool of state spaces has been obtained during the experiment, juxtaposing many different pairs (and even triples) of variables. The recognition of constraints and their illustration in state spaces has fulfilled two goals of this study. First, the state spaces *illustrate behavior* in terms of movement of vehicles, resource usage, and strategy and tactics in handling constraints. The participants' coping with constraints is illustrated in the state space by the movement of states towards or away from constraint lines. Review of the state spaces has identified trials with many crossings of constraint lines. In these trials, trucks frequently ran into difficulties and the fires burned out of control. In other state spaces, there were few or no crossings of constraint lines. This contrast indicates a large variety in the handling of constraints across teams and scenarios. Second, the study suggests that *representation design* in decision support systems and interfaces may usefully be based on the state space representation. In this study, information made available post-hoc in the state spaces suggests that the spaces give useful insight through the combination of essential variables and their behavior over time. Figure 9 recapitulates the state spaces in an abstract conceptual form.

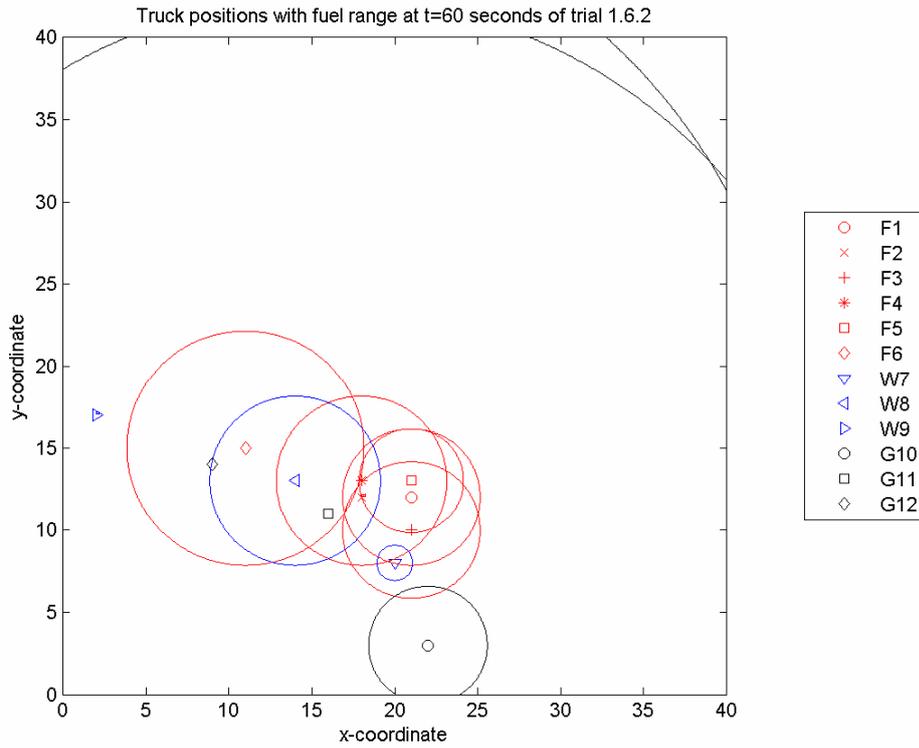


Figure 7 Map illustrating the constraint on fuel logistics at time  $t = 60$ , one minute into trial 1.6.2. All trucks are within range of at least one fuel truck.

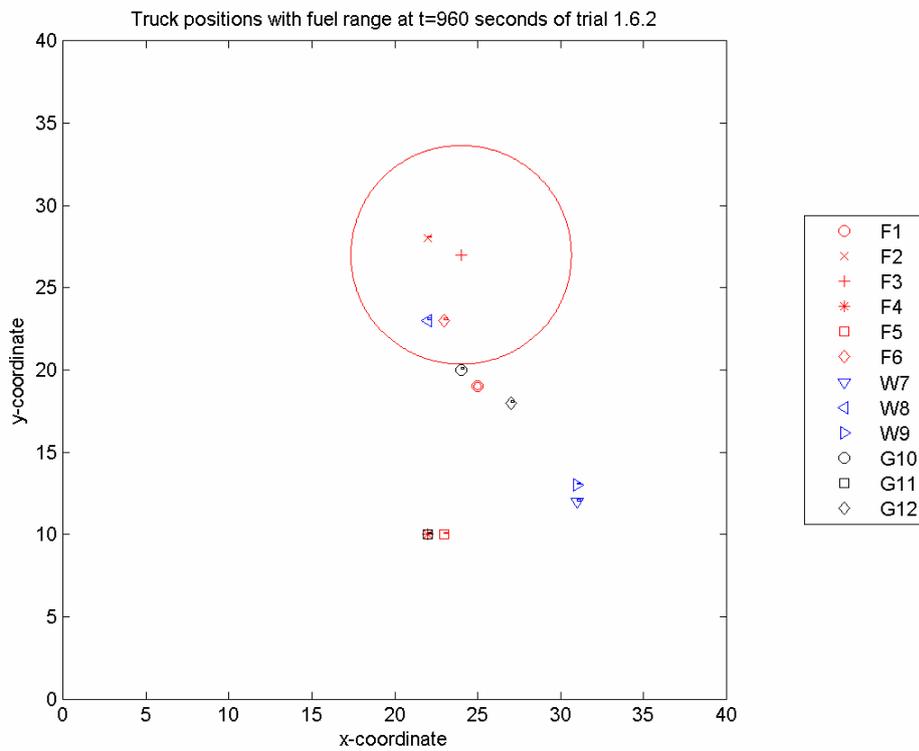


Figure 8 Map illustrating the constraint on fuel logistics at time  $t = 960$ , 16 minutes into the trial. No trucks can get refueled.

The state space on the left hand side of Figure 9 illustrates regions specifying differing opportunities for action. For states in the green region, fuel range is sufficient to bridge the distance. In the amber zone one cannot be certain if fuel range is sufficient, and the red region indicates that fuel is insufficient to bridge the distance. The first example of a state space (Figure 5) is a concrete example of this kind of state space. The state space on the right of Figure 9 illustrates a more gradual specification of constraint, and more interlaced regions with opportunities for action. A concrete example of such a state space is Figure 6 which juxtaposes the time available for an activity with the time needed for a related activity. In such cases the combination and juxtaposition of constraints may not be as rigid, but can nevertheless be relevant in a goal-directed context. Other shapes of regions of opportunities for action with more constraint lines combined may be identified in the future, as they have been reported in the literature in other domains. The combination of essential variables into one visualization can be highly meaningful and useful.

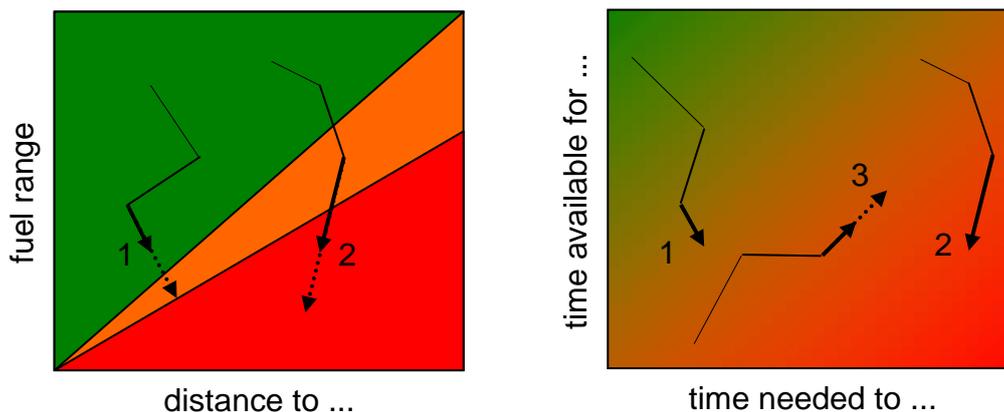


Figure 9 Conceptual idea of illustrating behavior and representing constraints in state spaces. Arrows represent state transitions over time. Thickness of the solid lines illustrates recency of state transition. Dotted arrows illustrate possible projections of future behavior.

State spaces form a very powerful representation. They can combine essential variables to form meaningful ensembles that are relevant to the goals of command and control. A value of an essential variable may not be informative in itself, but in the context of other variables and commander's goals, meaning is added. Thus, the visually lucid nature of state spaces lets their observer interpret the value of essential variables in a goal-directed context.

Automation or "intelligent" systems that aim to take over the task of interpreting data from process control operators have proven brittle and precarious in the face of unanticipated events, as discussed above. State spaces rearrange data in a way that facilitates interpretation and extraction of meaning, leaving the cognitive work and interpretation of data to the human controller, and thus avoiding the pitfalls of automation.

To control a process means to know the history of the process, its current situation or state, and to be able to anticipate and affect future states (Hollnagel & Woods, 2005). State spaces address the behavior of essential variables in the process to be controlled through the recent history of the process, the current state of the process. They enable extrapolation of the past and present to anticipate future behavior. Process data represented in state spaces may therefore enhance the controller's ability to control the process. Thus, the availability of representations of variables based on state space visualization may enhance control significantly for commanders in network-based command and control structures. Future network-based environments are expected to display a high reliance on data communication and presentation (Brehmer & Sundin, 2000; Cebrowski & Garstka, 1998). These settings are foreseen to pose high demands on the commander of the future to make sense of massive

amounts of data. State spaces offer a viable alternative to displays of raw data. Instead of presenting just raw data, state spaces offer a context for values of essential variables that may help commanders make sense of the dynamic situation. Instead of algorithms of “intelligent” decision support systems, state spaces let the commanders interpret data themselves, keeping them “in the loop”.

It is important to discuss the assumptions that the current work is based on. The first set of assumptions involves those inherited from the experimental platform, C3Fire. Although C3Fire captures many essential factors and properties of network-based command and control systems, there are some obvious differences between C3Fire and command and control environments. The C3Fire environment is clearly defined in terms of states of variables and their values. It is entirely deterministic, and can be called a closed-loop system. In command and control settings, the variables that play a role in the control of complex systems may not always be clearly identifiable. Information about the values of these variables may be uncertain, unreliable or incomplete. The utility of the method of recognizing constraints is not necessarily affected by these admittedly large differences. The state space representation does not interpret data; it merely enables human operators to interpret data and find meaningful patterns and relationships. The method does not assume that data is certain, reliable, or complete; it merely re-arranges data in a manner that makes the constraints on opportunities for action obvious.

In real command-and-control applications, as in C3Fire, state spaces can only juxtapose variables that are readily identifiable and observable. When variables are not available, the state spaces will not generate inappropriate depictions of the process, unlike much automation that has an interpretative role in supporting the operator. Because state spaces do not fabricate interpretations, unreliable or uncertain data are not a threat to the operators’ assessment of the situation.

The representations developed here are meant to be incorporated in the network-based visions of distributed collaborative command and control. It therefore assumes the availability of data by means of a broadcasting structure of communication, where each information-providing entity sends information to a network, and entities connected to this network have access to this information, listening in on the information they need to fulfill their tasks. Although the method surely can be applied to a certain extent to situations where operators have to actively seek information, the information considered here is assumed to be readily available. The set of essential variables should therefore be identified either at the design stage of support system development, or in real-time by the commanders given a flexible representation of data. Issues of network bandwidth, data filtering, and communication technology should therefore be ascribed to the idea of the network structure of command and control, and not to the constraint recognition method outlined here.

These considerations lead to a number of directions for future research. Regarding the description of behavior in terms of state spaces, it seems fruitful to continue analyzing data from a variety of command and control settings in terms of state spaces and to develop a more general method of representing constraints and their implications for behavior.

Regarding the design of representations of constraints and behavior in state spaces, much is unexplored in the details of the representations. As was discussed above, the approach supports the construction of understanding in terms of what has happened, what is happening, what may happen next, and which actions one can or cannot do. This promises to be a powerful approach. Questions concerning exactly which essential variables to present in order to improve control remain unexplored and have to be tailored to the specifics of the domain.

The theories outlined in earlier sections of this paper point us in several directions for further research. The recognition-primed decision model leads us to the importance of situation assessment and the recognition of the situation as being familiar and related to earlier experiences. Dynamic decision making emphasizes the relationship between goals and opportunities for action. The method for constraint recognition addresses these issues directly by identifying the goals, the means to achieve them, and the constraints on the opportunities for action as identified by examining the relationships

between essential variables. Further investigations into recent theories of command and control in relation to the constraint recognition method seem fruitful, such as the network-based paradigm described earlier in this thesis, and newly developed dynamic cybernetic models that focus on control (Brehmer, 2005).

Also, we are currently extending the scope of the method from emergency management to command and control where an adversary takes the role of the process to be controlled, as in traditional military settings, for example through the use of the dynamic wargame for experiments DKE (see Kuylenstierna *et al.*, 2004, for other experiments with this simulation).

In cognitive systems engineering some goals of representation design are to reduce the time needed for evaluation and the time needed for the selection of action, as well as to facilitate simultaneous compensatory and anticipatory control (Hollnagel & Woods, 2005). We claim that the state space representation has the potential to do just that. As is discussed in the paper, the state space approach supports the construction of understanding in terms of what has happened, what is happening, what may happen next, and which actions one can or cannot do. This promises to be a powerful approach to support the maintaining and regaining of control, since control support means support for compensatory and anticipatory goal-directed action.

In many time-critical systems in dynamic complex domains, a balance needs to be found between safety and efficiency (Hollnagel, 1999). The notion of control encompasses both of these aspects of system behavior, and research approaches directed at the facilitation of control such as the recognition of constraints may improve our understanding of establishing both safety and efficiency to a satisfactory extent. Instead of the brittleness of many earlier approaches to ‘decision support’, the constraint-based approach may bring design a little closer to enable systems to behave in a resilient manner. Enhanced conditions for anticipation, preparation, and recovery may facilitate resilience (Hollnagel *et al.*, 2006), a concept that addresses both safety and efficiency.

## Acknowledgement

The Swedish Defense Material Administration is gratefully acknowledged for supporting this research.

## References

- Adelson, M. (1961). Human decisions in command control centers. *Annals of the New York Academy of Sciences*, 89, 726-731.
- Ashby, W. R. (1956). *An introduction to cybernetics* (Internet (1999) ed.). London: Chapman & Hall. <http://pcp.vub.ac.be/books/IntroCyb.pdf>.
- Ashby, W. R. (1960). *Design for a brain: The origin of adaptive behavior* (2nd (revised) ed.). London: Chapman & Hall Ltd.
- Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19(6), 775-779.
- Brehmer, B. (1992). Dynamic decision making: Human control of complex systems. *Acta Psychologica*, 81, 211-241.
- Brehmer, B. (2005). *The dynamic OODA loop: Amalgamating Boyd's OODA loop and the cybernetic approach to command and control*. Paper presented at the 10th International Command and Control Research and Technology Symposium (ICCRTS) - The future of C2, McLean, VA.
- Brehmer, B., & Allard, R. (1991). Real-time dynamic decision making: Effects of task complexity and feedback delays. In J. Rasmussen, B. Brehmer & J. Leplat (Eds.), *Distributed decision making: Cognitive models for cooperative work*. Chichester: Wiley.

- Brehmer, B., & Dörner, D. (1993). Experiments with computer-simulated microworlds: Escaping both the narrow straits of the laboratory and the deep blue sea of the field study. *Computers in Human Behavior*, 9(2-3), 171-184.
- Brehmer, B., & Sundin, C. (2000). Command and control in network-centric warfare. In C. Sundin & H. Friman (Eds.), *ROLF 2010 - the way ahead and the first step* (pp. 45-54). Stockholm: The Swedish National Defence College.
- Cebrowski, A. K., & Garstka, J. J. (1998). Network-centric warfare: Its origin and future. *US Naval Institute Proceedings, January*, 28-35.
- Chapman, R., Smith, P. J., Billings, C. E., McCoy, C. E., & Heintz Obradovich, J. (2001). *Collaborative constraint propagation as a planning strategy in the national airspace system*. Paper presented at the 2001 Annual Meeting of the IEEE Society on Systems, Man and Cybernetics, Tucson, AZ.
- Checkland, P. (1981). *Systems thinking, systems practice*. Chichester: John Wiley & Sons.
- De Jong, J. J., & Köster, E. P. (1974). The human operator in the computer-controlled refinery. In E. Edwards & F. P. Lees (Eds.), *The human operator in process control* (pp. 196-205). London: Taylor & Francis Ltd.
- Dekker, S. W. A., & Woods, D. D. (1999). To intervene or not to intervene: The dilemma of management by exception. *Cognition, Technology & Work*, 1(2), 86-96.
- Dekker, S. W. A., & Woods, D. D. (2002). Maba-maba or abracadabra? Progress on human-automation co-ordination. *Cognition, Technology & Work*, 4(4), 240-244.
- Flach, J. M., Smith, M. R. H., Stanard, T., & Dittman, S. M. (2003). Collisions: Getting them under control. In H. Hecht & G. J. P. Savelsbergh (Eds.), *Theories of time to contact* (pp. 67-91). North-Holland: Elsevier.
- Funke, J. (2001). Dynamic systems as tools for analysing human judgement. *Thinking and Reasoning*, 7(1), 69-89.
- Gibson, J. J. (1986). *The ecological approach to visual perception*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Granlund, R. (2002). *Monitoring distributed teamwork training*. Unpublished PhD Thesis, Linköpings universitet, Linköping.
- Gray, W. D. (2002). Simulated task environments: The role of high-fidelity simulations, scaled worlds, synthetic environments, and laboratory tasks in basic and applied cognitive research. *Cognitive Science Quarterly*, 2, 205-227.
- Hollan, J., Hutchins, E., & Kirsch, D. (2000). Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction*, 7(2), 174-196.
- Hollnagel, E. (1986). Cognitive systems performance analysis. In E. Hollnagel, G. Mancini & D. D. Woods (Eds.), *Intelligent decision support in process environments* (pp. 211-226). Berlin: Springer-Verlag.
- Hollnagel, E. (1993). *Human reliability analysis: Context and control*. London; San Diego, CA: Academic Press.
- Hollnagel, E. (1999). From function allocation to function congruence. In S. Dekker & E. Hollnagel (Eds.), *Coping with computers in the cockpit* (pp. 29-53). Aldershot: Ashgate.
- Hollnagel, E. (2003). Cognition as control: A pragmatic approach to the modelling of joint cognitive systems. *IEEE Transactions on Systems, Man and Cybernetics, in press*.
- Hollnagel, E. (2004). *Barriers and accident prevention*. Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.
- Hollnagel, E., & Bye, A. (2000). Principles for modelling function allocation. *International Journal of Human-Computer Studies*, 52(2).

- Hollnagel, E., & Woods, D. D. (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man-Machine Studies*, 18, 583-600.
- Hollnagel, E., & Woods, D. D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering*. Boca Raton, FL: CRC Press.
- Hollnagel, E., Woods, D. D., & Leveson, N. (Eds.). (2006). *Resilience engineering: Concepts and precepts*. Ashgate.
- Hutchins, E. (1991a). Organizing work by adaptation. *Organization Science*, 2(1), 14-39.
- Hutchins, E. (1991b). The social organization of distributed cognition. In L. Resnick, J. Levine & S. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 283-307). Washington, D.C: APA Press.
- Klein, G. A., & Calderwood, R. (1991). Decision models: Some lessons from the field. *IEEE Transactions on Systems, Man and Cybernetics*, 21(5), 1018-1026.
- Klein, G. A., Ross, K. G., Moon, B. M., Klein, D. E., Hoffman, R. R., & Hollnagel, E. (2003). Macro-cognition. *IEEE Intelligent Systems*, 18(3), 81-85.
- Knecht, W. R., & Smith, K. (2001). The manoeuvre space: A new aid to aircraft tactical separation. In D. Harris (Ed.), *Engineering psychology and cognitive ergonomics* (Vol. 5, pp. 197-202). Aldershot: Ashgate.
- Kuylensstierna, J., Rydmark, J., & Sandström, H. (2004). *Some research results obtained with DKE: A dynamic war-game for experiments*. Paper presented at the 9<sup>th</sup> International Command and Control Research and Technology Symposium (ICCRTS). Copenhagen, Denmark.
- Leveson, N. G. (2004). A new accident model for engineering safer systems. *Safety Science*, 42, 237-270.
- Lind, M., & Larsen, M. N. (1995). Planning support and the intentionality of dynamic environments. In J.-M. Hoc, P. C. Cacciabue & E. Hollnagel (Eds.), *Expertise and technology: Cognition & human-computer cooperation* (pp. 255-278). Hillsdale, N.J.: Lawrence Erlbaum Associates.
- Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the structure of behavior*. New York: Holt, Rinehart and Winston, Inc.
- Neisser, U. (1976). *Cognition and reality: Principles and implications of cognitive psychology*. San Francisco, CA: W. H. Freeman and Company.
- Newell, A., & Simon, H. A. (1961). GPS: A program that simulates human thought. In H. Billings (Ed.), *Lernende automaten* (pp. 109-124). München: R. Oldenbourg.
- Norman, D. A. (1998). *The design of everyday things*. London, England: MIT Press.
- Persson, P.-A. (1997). *Toward a grounded theory for support of command and control in military coalitions*. Unpublished Licentiate Thesis, Linköpings universitet, Linköping, Sweden.
- Port, R. F., & Van Gelder, T. (Eds.). (1995). *Mind as motion: Explorations in the dynamics of cognition*. Cambridge, MA: MIT Press.
- Reitman, W., R. (1964). Heuristic decision procedures, open constraints, and the structure of ill-defined problems. In M. W. Shelly II & G. L. Bryan (Eds.), *Human judgments and optimality* (pp. 282-315). New York: John Wiley & Sons, Inc.
- Roth, E. M., Malin, J. T., & Schreckenghost, D. L. (1997). Paradigms for intelligent interface design. In M. Helander, T. K. Landauer & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., pp. 1177-1201). Amsterdam: Elsevier Science.
- Smith, K. (1997). *How currency traders think about the spot market's thinking*. Paper presented at the Cognitive Science Society 19th Annual Conference.
- Smith, K. (2001). Incompatible goals, uncertain information, and conflicting incentives: The dispatch dilemma. *Human Factors and Aerospace Safety*, 1(4), 361-381.

- Smith, W., & Dowell, J. (2000). A case study of co-ordinative decision-making in disaster management. *Ergonomics*, 43(8), 1153-1166.
- Stappers, P. J., & Flach, J. M. (2004). *Visualizing cognitive systems: Getting past block diagrams*. Paper presented at the IEEE International Conference on Systems, Man & Cybernetics, The Hague, The Netherlands.
- Vicente, K. J. (1999). *Cognitive work analysis: Towards safe, productive, and healthy computer-based work*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4), 589-606.
- Woltjer, R. (2005). *On how constraints shape action*. Unpublished Licentiate thesis, Linköpings universitet, Linköping, Sweden.
- Woltjer, R., & Smith, K. (2005). *Constraint recognition in a fire-fighting microworld study*. In Woltjer, R. (2005). Unpublished manuscript, Linköping.
- Von Neumann, J., & Morgenstern, O. (1953). *Theory of games and economic behavior* (3rd ed.). Princeton, NJ: Princeton University Press.
- Woods, D. D. (1986). Paradigms for intelligent decision support. In E. Hollnagel, G. Mancini & D. D. Woods (Eds.), *Intelligent decision support in process environments* (pp. 153-173). Berlin: Springer-Verlag.
- Woods, D. D. (1995). Toward a theoretical base for representation design in the computer medium: Ecological perception and aiding human cognition. In J. Flach, P. Hancock, J. Caird & K. Vicente (Eds.), *Global perspectives on the ecology of human-machine systems* (pp. 157-188). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.