Fall 2012

Level of Agreement in the Mental Models of Human Factors Practitioners and Systems Engineers Working in Collaborative Teams

Jerry A. Gordon
Embry-Riddle Aeronautical University - Daytona Beach

Scholarly Commons Citation
Gordon, Jerry A., "Level of Agreement in the Mental Models of Human Factors Practitioners and Systems Engineers Working in Collaborative Teams" (2012). *Dissertations and Theses*. 74.
https://commons.erau.edu/edt/74

Level of Agreement in the Mental Models of Human Factors Practitioners and Systems Engineers working in Collaborative Teams

by

Jerry A Gordon
B.S. Marine Engineering, United States Naval Academy 1991

A Thesis Submitted to the
Department of Human Factors and Systems
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Human Factors and Systems

Embry-Riddle Aeronautical University
Daytona Beach, Florida
Fall 2012

Level of Agreement in the Mental Models of Human Factors Practitioners and Systems Engineers working in Collaborative Teams

by

Jerry A Gordon

This thesis was prepared under the direction of the candidate's thesis committee chair, Beth Blickensderfer , Ph.D., Department of Human Factors & Systems, and has been approved by the members of the thesis committee. This thesis was submitted to the department of Human Factors & Systems and has been accepted in partial fulfillment of the requirements for the degree of Master of Science in Human Factors & Systems.

THESIS COMMITTEE:

Beth Blickensderfer, Ph.D., Chair

Kelly Neville, Ph.D., Member

John J. Burns, Ph.D., Member

MS Human Factors & Systems Program Coordinator

Department Chair, Department of Human Factors & Systems

Associate Vice-President of Academic Affairs

Table of Contents

List of Tables

List of Figures

Abstract

Emerging research in complexity science recognizes traditional techniques for engineering systems do not always work for complex systems. Designing complex systems requires individuals to have knowledge of engineering as well as human performance. To this end, design efforts rely often on multi-disciplinary teams. While any two members of a design team may view the system design problem in vastly different manners, this study sought to identify a possible *systemic* effect on approach by the differing education and experience obtained by social practitioners, represented by human factors, and technical practitioners, represented by systems engineers. It further examined the impact of the complexity of the designed system designed on this systemic effect; in this case, two systems associated with unmanned aircraft systems (UAS). This study relied on measurement of individual mental models, using a graphical brainstorming tool to capture functional decompositions, argued as representing the problem domain component of an individual mental model. This study compared individual functional decomposition models against an average model composed from the same educational specialty, and from an average model composed from the opposite educational specialty. Participants developed models for a simple/closed problem and an open/complex problem. The researcher conducted a repeated measures multivariate analysis of variance on the effects of domain, problem type and the interaction between the two, as well as with interactions with educational specialty. The results indicated higher agreement among mental models when individuals were compared to the average model from their same specialty, that more agreement in mental models occurred in relation to the simple/closed problem than in relation to the open/complex problem, and that open/complex problems can exacerbate the level of mental model dis-agreement among team members with different educational backgrounds.

Introduction

In recent years, Human Factors practitioners are increasingly called to participate in complex system development early in the development process. The author has participated in over a dozen of these large programs during his 20+-year career in engineering. There are certainly examples of systems where joint participation of practitioners from both human and technical domains, has improved the overall utility, cost effectiveness and performance of these systems (Militello, Dominguez, Lintern & Klein, 2010). However, there are also other programs where this joint participation has been more difficult, and the ultimate enterprise resulted in failure (e.g. Constellation, Future Combat Systems). While the reasons for failure are not always clearly understood, many program failures are blamed on bad requirements management. The author has observed, and there is research evidence to suggest, that requirements failures are among, other things, a product of stakeholders' failure to agree upon the scope and direction of the project at each stage in a timeframe that presents acceptable cost and schedule to the sponsors (Johnson & Holloway, 2006). Johnson and Holloway (2006) called this "inadequate conflict management."

Practitioners of technical (e.g. systems and software engineering) and social (e.g. human factors) domains, have different education and experience, thus it is likely, that they view and more importantly, describe, the purpose or goal of the design problem in different ways. In turn, this difference of perspective and terminology increases with the complexity of the system problem.

The purpose of this research is to examine the differences in perspectives between technical practitioners and social domain practitioners. To accomplish this, the research will leverage complexity theory, cognitive systems engineering, and mental model theory. It shall argue that graphical models of a system developed by technical and human factors practitioners are equivalent to part of the individual mental model associated with domain knowledge as it relates to the causal understanding of goals-means relationships. It will discuss various graphical methods used within those domains, and demonstrate that a hierarchical graph of system goals and behaviors (i.e. functions) resulting from a brainstorming session is semantically equivalent to the *functional flow decomposition* used by technical practitioners and the *functional abstraction hierarchy* used by social practitioners when describing systems in a common setting. As such, it will use a hierarchically arranged brainstorming tool to measure and compare the mental model components within and between groups of like practitioners.

Complex Systems

Complexity is a term often used in engineering, but there is little agreement as the exact definition (Vicente, 1999). Similarly, there is recognition that there is a class of *complex* systems, which require qualitatively different approaches to engineering than traditional systems engineering (Minai, Braha & Bar-Yam, 2006). Additionally, there is a great deal of discussion on identifying specific criteria to classify a system as complex. A working definition is that complexity is the uncertainty involved in achieving (which may include “proper understanding of”) stakeholder requirements (Suh, 2001). There are a number of similar definitions of complexity used by systems engineers that all stem from interpretations of Shannon’s information theory (Buede, 2000). Within the Human Factors community, Vicente defines complexity as the ability to predict the behavior of a system. Combined with formal Bayesian probabilities, Vicente’s definition can provide a more mathematical definition that allows for a lack of stated requirements. Woods and Hollnagel address it from the perspective of human coping, referencing the cybernetics “law of requisite variety.” This law states that the number and range of options to complete the work must be at least equivalent to the number and range of constraints imposed by the work environment. All of these authors define complexity in terms of *information content* (Suh, 2001); the numbers of variables present in a system and must be known to predict its behavior; and whether that behavior can be controlled to desirable ends.

The International Council on Systems Engineering (INCOSE) Complex Systems Working Group (Sheard & Mostashari, 2009) proposed a more practical definition of complexity. Table 1 articulates the INCOSE definition, and the attributes of the problem statements used in this research to establish a "complex" problem.

Table 1
Complex systems characteristics and interpretations for validating a problem as "complex."

INCOSE Definition	Interpretation for Experiment
Composed of autonomous components	High level of automated functions; boundaries between system and environment are not always distinct
Self-organizing	May be deployed in a number of different and undefined configurations
Emergent behavior - non linear	Small changes in requirements or external constraints (rules of engagement, communication frequencies, air traffic) have large impacts on system behavior. Can generate disruptive events
Adapt to environment	Can be reconfigured in real time. Can recover from disruptive events
Increase in complexity over time	Can add additional components in real time

Note: Contrast with a large system, or a complicated system possessing a large number of parts. In both cases, they may not be complex if they do not exhibit these characteristics.

Thus, systems engineering, as the concept is currently understood as a reductionist approach (Sheard & Mostashari, 2009), encounters new dimensions of difficulty from complexity. This is due to the open nature of modern systems, which are constantly undergoing evolution, expecting composition into multiple *systems of systems,* which include many components outside the direct control of the designers or operators. Norman and Kuras (Norman & Kuras, 2006) write that these open systems are best viewed as an *enterprise* that has several characteristics that distinguish it from a traditional system.

Sheard adds that the words “enterprise” and “architecture” have now become semantically overloaded and inadequate to discuss the implications of developing complex systems. In addition to restatements of the requirements of table 1, she adds that complex systems display a fractal structure, where the low level components cannot easily be discovered by analysis of macro level structures, although they do exhibit repeating patterns on ever increasing scales.

The next few subsections will discuss specific examples of the complex systems characteristics presented in Table 1. Following this discussion is the introduction of the candidate problem domain for this research, Unmanned Aircraft Systems (UAS). This section will conclude with analysis of UAS as satisfying the requirements of complexity.

Automation

The first attribute of complex systems is the high level of automation. Complexity in modern systems is driven by automation and in particular by automation software. The impact of advances in automation on the human components in systems is a well-researched topic (Sheridan, & Parasuraman, 2006). The original work done in the 1980’s (Bizantz & Burns, 2009), coincided with the rise of automation and computer based manufacturing and other work.

Automation arose to make life easier for humans. While the potential downsides of technology have been the subject of authors since “Rossum’s Universal Robots” at the dawn of the 20th century, it was generally thought that automation would make for less work and it certainly has reduced the need for human muscle (Woods & Hollnagel, 2006). Subsequent to actually fielding automated systems, however, researchers discovered that the nature of the work had simply changed. Automation came with its own challenges, such as *automation surprises* (Woods, Patterson, & Roth, 1998). Far from eliminating humans in the work environment, automation created a completely new set of challenges for designing work environments for which humans could succeed (Parasuraman & Wickens, 2008).

Automation exists along a continuum from fully manual to fully autonomous. In 1978, Sheridan and Verplank (as cited in Endlsey & Kaber, 1999) presented the following 10 level taxonomy for describing autonomy

1) Human does the whole job up to the point of turning it over to the computer to implement;
2) Computer helps by determining the options;
3) Computer helps to determine options and suggests one, which human need not follow;
4) Computer selects action and human may or may not do it;
5) Computer selects action and implements it if human approves;
6) Computer selects action, informs human in plenty of time to stop it;
7) Computer does whole job and necessarily tells human what it did;
8) Computer does whole job and tells human what it did only if human requests notification
9) Computer does whole job and decides what the human should be told; and
10) Computer does the whole job if it decides it should be done, and if so, tells human, if it decides that the human should be told

In practice, automation has advanced from the information support and repetitive motion of its early days (Sheridan & Parasuraman, 2006) to complex decision and action automation with *fully automatic control*. Robots are now making decisions and executing actions, and only then informing their human monitors (Woods & Hollnagel, 2006). In some cases, aircraft mishaps have occurred when the human monitors were not informed and made inappropriate actions based on an erroneous interpretation of aircraft state. An example is the lack of engine-off alert on the Predator B crash in Arizona. The receiving Ground Control Station (GCS) gave the aircraft command for control hand off, with the engine shutoff switch left in the cutoff position following the landing and recovery of a different aircraft. The aircraft, by way of the automation designers, assumed an engine off signal would only come from the operators with deliberate intent, and did not put in any kind of state warning or callback to affirm the command (National Transportation Safety Board, 2006).

While automation is a significant contributor to complexity, automation is not enough to make a system complex by itself. The next subsection continues with the characteristics of self-organizing systems – another critical component in complexity.

Self-Organizing Systems

Another characteristic of complex systems is that they are self-organizing. Self-organization is an emergent property of systems that have both *connectivity and interdependence*; autonomy and loose-coupled connections. They are capable of dynamically forming relationships and structures to pursue mutual goals (when they detect or agree that their goals are mutual), and departing, or reorganizing when the utility of that structure to increase goal satisfaction is no longer optimal (Mitleton-Kelly, 2003). Classic examples of self-organizing systems are s*warming behaviors*, where independent actors like software agents, “nanobots” or autonomous vehicles, can gather into a flock or swarm (the name taken from biology to describe birds or insects working together), to perform a task. An example which also provides a cautionary tale is the May 5, 2011 “flash crash” of the New York Stock Exchange, where stock monitor software “bots” swarmed on bad news and executed a massive sell off automatically, with the Dow Jones Industrial Average losing over 1,000 points in a few minutes. A more positive example is found in the advances in targeted cancer therapies using gold nano-particles (that travel to and coalesce on cancerous tissue due to inherent connective properties between the gold particles and the cancerous cells) to “cook” tumors using radiation that is preferentially absorbed in the gold (Using gold nanoparticles, 2010).

The reason that self-organizing behaviors make a system complex is that by definition, the final form of the system is difficult to predict. When the form is partly unknown, the functions performed by the system are also likely in flux. Complicating this prediction is that complex systems often present vastly transformative effects from small changes in their form, as discussed in the next subsection.

Non-Linear Emergent Behaviors

Complex systems also exhibit *non-linearity*, which typically means that the system responses, which may include functions of the linear “$\mathbf{y}=m\mathbf{x}+b$” variety, do not obey the rule of superposition; where the final output function cannot be described by the addition of two or more different inputs functions. Non-linearity is described in complexity science by the behavioral effect of having large variations in output values for correspondingly small changes in inputs. In particular, output cannot be predictably described as a function of input (Minai, et al, 2006). Chaotic systems exhibit a behavior that, even when all factors are measured (even the practically non-measurable ones), the resultant behavior cannot be distinguished from random chance. Edge systems are those that are complex and occasionally exhibit chaotic behaviors. A classic example is the internet, as Xerox’s Palo Alto Research Center (PARC) would never have predicted the current information driven political revolutions in the Middle East when they established the protocols that enable the internet. Since then, a series of seemingly simple inventions, such as web logging (i.e. “blogging”) and smart phones has enabled a worldwide phenomenon with far ranging consequences that will likely last for decades.

The last characteristic of complexity as defined by INCOSE is the ability to adapt, often in real time, to environmental pressures. The next subsection discusses adaptability

Adaptability

Research into complexity theory has leveraged the life sciences to understand adaptability, similar to living creatures, complex systems and organizations evolve over time in response to external environmental pressures (Rouse, 2007). These pressures can include technological or regulatory change, as well as a change in the intended purpose of the system – as its operating environment changes over time (Bartolemei, Hastings, De Neufille & Rhodes, 2012).

Moreover, these systems have the power to change themselves, either by the operators selecting new goals for the system in real time as it operates, or as a “system of systems” where the components of the system enjoy significant autonomy, and different combinations of components may aggregate for different missions in a number of, often unforeseeable, configurations.

An example of adaptability is the additional missions assigned the P-3 Neptune Maritime Patrol aircraft, and its successor the P-8. While initially fielded to detect and engage Soviet ballistic missiles submarines (SSBN), it has evolved with new sensors and innovative uses for older sensors to provide a multitude of new capabilities (Gordon, Burns, Sheehan, Ricci, & Pharmer, 2005). These include surveillance of surface craft for enforcement of blockades and customs, searching for mines, providing third part targeting for cruise missiles and conducting search and rescue activities. None of these new missions could have been foreseen, any more than could the dissolution of an imminent Soviet SSBN threat. The aircraft, however, has proved to be adaptable and lived long past its original intended life to provide excellent capability.

In summary, complex systems have a variety of characteristics including automation, self-organization, non-linearity and adaptability. Unmanned Aircraft Systems (UAS) exemplify a number of these characteristics. The next section will discuss UAS.

Unmanned Aircraft Systems (UAS)

Unmanned Aircraft Systems (UAS) are composed of remotely controlled aircraft with varying levels of onboard automation (Sheridan & Parasuraman, 2006). Those controllers are human operators, who may utilize varying levels of automation in their GCS as well as on the aircraft itself (Cummings, Kirschbaum, Sulmistras, & Platts, 2006). Moreover, UAS are designed to carry a number of different payloads to perform a variety of missions, and they have multiple levels of control, which can be passed between different human users and levels of automation, in a large number of ultimate configurations. During the past decade of operations in the Middle East, UAS have gone from novelty item to a principal warfighting platform and usage is expected only to grow in the future. Thus, they provide a useful domain for analysis of complexity. Table 2 lists examples of UAS and complexity attributes, which will define the Automated Mission Control Software (AMCS) problem statement for this research.

Table 2

Problem characteristics of the complex Automated Mission Control System (AMCS)

Attribute of Complexity	Open Problem Case as “Complex”
Autonomy	Autopilot automatically follows navigation plans and fly aircraft. UAS operators, ATC, FDC, supported commands all act independently and in concert.
Self-Organizing	Chat functions allow UAS to dynamically support multiple customers in ad hoc, supporting/supported organizations as defined by command staffs
Nonlinear Emergent Behaviors	Impact of Drone warfare on battlefield ethics and law
Adaptability and Composability	STANAG control handoff, Mission Packages

Note: Air Traffic Control (ATC), Fire Direction Center (FDC), Unmanned Aircraft System (UAS), Standard North American Treaty Organization Agreement (STANAG) are terms used in the military unmanned aircraft domain.

UAS provide an opportunity to both define a more traditional problem, as well as a problem exhibiting all the characteristics of complexity outlined above. First consider a *closed* or simple (as opposed to complex) problem, such as develop mission monitoring software. The attributes of this closed problem contrast with the requirements of complexity as described in Table 3.

Table 3
Mission Control Software characteristics contrasted with complex systems attributes

Attribute of Complexity	Closed Problem Case as "Simple"
Autonomy	The system can execute automatic tasks, but only as directed by the operator controlling the aircraft, and the automation provides status updates
Self-Organizing	The mission control software is installed in the GCS and operates with the aircraft of a known configuration. Product has a well-defined boundary.
Nonlinear Emergent Behaviors	The desired behaviors are those required to support the phases of flight. Undesired behaviors are designed away during product development
Adaptability and Composability	The functions of the mission control software can de decomposed and allocated to human and machine functions, according to traditional systems engineering practices

Note: Complexity defined by International Council on Systems Engineering (INCOSE). Ground Control Stations (GCS) is where pilots remotely operate unmanned aircraft.

In terms of designing and developing complex systems, various methods have been proposed to address this complexity. One of these, cognitive systems engineering, arose from the human factors/psychology discipline, but is presented as a systems engineering method to address system design. As such, it represents the community of practice that seeks to join the expertise of both the social (human factors) and technical (systems engineering) domains. This will be discussed next.

Addressing Complexity in Systems Engineering

Cognitive Systems Engineering

Increasing levels of complexity in modern systems development has led to the formation of a number of disciplines and techniques to address this complexity. *Cognitive systems engineering* (CSE) is a set of techniques which draws on human factors methods and scientific principles with the intent of addressing human issues in the development of complex socio-technical systems directly through the requirements process (Militello, Dominguez, Lintern & Klein, 2010). CSE is a relatively new discipline, and it continues to vie for acceptance as part of the overall systems engineering process, although it has enjoyed some early successes (Madni, 2010). A key portion of the CSE process is the analysis of the human work involved in the operations of the socio-technical system designed.

CSE analysis techniques rely heavily on modeling, including various approaches for describing the nature and purpose of work, human performers of the work and information content of the work space. These techniques evolved from the family of cognitive task analysis (CTA) techniques developed in the 1980s by Jens Rasmussen and Kim Vicente to develop user interfaces and training systems, among other uses (Jonassen, Tessmer, & Hannum, 1999). One mechanism for modeling cognitive work is the goals/means decomposition, which describes a hierarchy of *why work is performed* tied to *how this work is performed* (Jonassen et al., 1999).

Evolving from classical human factors, which in turn evolved from information theory of Shannon and Weaver (Hollnagel &Woods, 2006), these CSE methods used cognition as a starting point, and then built a model of the world around it. This represents a hermeneutic or “brain in a jar” perspective, where reality is an experience as interpreted through *sensors* and *actuators*, and the job of the HCI specialist is to identify, design and arrange the right sensors and actors. Early methods developed in cognitive systems engineering used variations on the hermeneutic approach, such as decision centered design, which arose from work performed on analyzing the USS Vincennes 1988 Iranian Airbus incident (Militello et al., 2010). These methods focused on providing more intuitive interfaces to conduct taskwork, and as such focused on the human machine interface. The next section introduces a new perspective that sought to broaden the focus to a concept called Joint Cognitive Systems.

Joint Cognitive Systems

Fortunately, an alternative philosophy has evolved, leveraging ideas from sociology; that of an *ecological approach* (Hollnagel &Woods, 2006). It identifies an *ecology* of humans, both augmented with *prostheses* and utilizing *tools* as *artifacts*, which have been selected to provide *cognitive affordances*, or opportunities to accomplish work. The ecological approach thinks of humans and technologies in *Joint Cognitive Systems* (JCS). The result is a socio-technical network of shared and augmented cognition; defining, evaluating, planning and executing work towards achieving purpose, the goals of which are established both internally and externally. The technology may be prosthetic (where the artifact is an extension of the human – an example of a most extreme case, the “bionic” artificial limb) or a tool (with a pencil serving as a simple example). The ecological approach thinks of humans as embodying the technology to sense, interact and affect the world, thus the technology must be designed to mesh cognitively with the human. The intent of the JCS concept is to enable focus on the totality of the work environment as an emergent system, rather than a focus solely on the *human-technology dyad*, which in practice tends to shift engineering emphasis to human-machine interface (HMI) design.
Figure 1 depicts these contrasted perspectives.

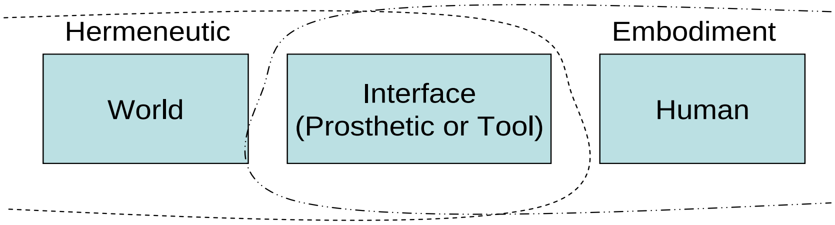

Figure 1. Different Task Analysis Perspectives of Work and the World. Adapted from Woods & Hollnagel, 2006. Embodiment views technology as extension of environment from the human out, whereas hermeneutic views the human out from the technology and environment.

The theoretical basis for the ecological methods lies in the activity theory of Vygotsky and others, developed as part of Soviet psychology (Woods & Hollnagel, 2006). They mandated an anthropocentric frame of reference as being the only one that matters, i.e. that technology had no meaning until actually used by someone, and that person's goals assumed primacy. In this view they speak of *cognitive affordances* as a property of elements in the environment, which are acted upon by the human users and their tools/prosthesis. Gibson presented his theory in the 1970's as an alternate to the Shannon-Weaver paradigm (Albrechtsen, Andersen, Bødker, & Pejtersen, 2001).

Several other practical approaches to conducting Cognitive Work Analyses have been designed from this ecological perspective (Hollnagel, 2003; Vicente, 1999; Lintern, 2009; and Potter, Elm, Roth, Gualtieri, & Easter, 2001). While Vicente considers cognitive task analysis (CTA) as a subspecialty of cognitive work analysis (CWA), CTA can be seen as assembling data from the hermeneutic approach and CWA as assembling data from the embodiment approach, although Lintern acknowledges that there is no general agreement on the difference within the CSE community.

One of the most recent approaches to CSE is the Applied Cognitive Work Analysis (ACWA). While grounded heavily in theory, the ACWA has evolved as a pragmatic answer to addressing specific challenges in ensuring that complex technical systems were developed with affordances*;* with work designed to “intuitively fit” its users (Hollnagel, 2003). A key construct of the ACWA process is the development of a “functional abstraction network” (FAN). This network describes the *goals-means decomposition* of the workspace, and becomes the scaffolding to build the cognitive and information demands and characteristics of the workspace. This goals-means decomposition is a behavioral or *functional* description of the system, from the perspective of the CSE/Human Factors practitioners developing it.

CSE and related HF techniques are going to be essential to the proper development, fielding, operations, and maintenance of complex systems such as UAS in the future. However, this will require CSE to proceed from its current “initial enthusiasm” (Madni, 2010) to a mature, repeatable discipline with demonstrated Return on Investment (ROI). Among other things, this is predicated on a proper understanding of the characteristics of a system, which will benefit properly from the application of CSE. It is also predicated on understanding the characteristics of system development efforts that affect communication in the large, multi-functional distributed teams that are often used to develop Defense and Aerospace systems.

CSE arose in part to address the impact of complexity on functional analysis and allocation. The human factors view of the functional allocation problem started with the “Machines Are Better At/Humans Are Better At” approach that substituted machine tasking for human tasking. It has migrated to recognition that simple substitution is insufficient, and allocation on that basis alone creates its own problems, such as automation surprises (Woods & Hollnagel, 2006). For earlier methods such as human centered design (Hollnagel, 2003), the technical and human factors perspectives still had the human machine interface (HMI) as a common point of reference, and it is reasonable to assume that the different groups could use this common point to maintain consistent mental models regarding the nature of the design; what they were building and why. The description of a JCS, which takes a qualitatively different approach to understanding the nature of the system, still has the HMI, but it is a conclusion rather than a starting point.

Vicente points out, however, that the cognitivist perspective tends to focus on the HMI, since it is here that the connections between the biological sensors and actuators and the rest of the world are located. As discussed previously, this provides for oversimplification of the CTA representation of the rest of the man-machine system. However, the author believes that the success realized in applying cognitivist CTA methods, in such approaches as *human centered design* (Vicente, 1999) is due in part to the relatively closed and fixed nature of the systems considered. This is not to say those systems weren't challenging, or even apparently complex, but rather the analysts could adopt a set of conventions and simplifications that could be understood by the various stakeholders *within* the timeframe of a tolerable task-artifact cycle. Additionally, it may have been true that most of the behaviors present in the rest of the man machine system had a direct analog on the HMI, and the description of the HMI from a cognitive perspective sufficiently complete to facilitate this understanding.

In the general case, however, there is much more to the complete man-machine system than the humans and their HMI. Addressing the complexity introduced by these other factors is a task for which the CSE and the concept of the JCS are ideally suited. A class of problems called *socio-technical systems* has been defined to describe organizations which include both significant technological and social aspects, which must be understood in detail to really understand the behavior of the organization (Osorio, Dori & Sussman, 2011). Socio technical systems are presented in the next subsection.

Complex Socio-Technical Systems

Socio-technical systems include emergent behaviors arising from the use of social networks and impact of cultural biases. Designers cannot understand the ultimate form and function of these systems without accounting for these additional factors. Thus, there is more to the human portion than individual cognition. Woods and Hollnagel call this “distributed cognition.” As discussed previously, there is more to the machine aspect than an HMI and the underlying data. Advances in automation have led to a great deal of modern system functionality executed with no explicit direction from the human operators. Thus, there is a lot to the *man machine system* that exists in the machine space, for which man has no direct mechanism to interact.

In addition to the human shared cognition and the pure automation of the machine, there are organizational aspects which exist above the humans and technology to describe the system. These organizational aspects include shared culture and sense of purpose, as well as institutional knowledge that shapes the perceptions of the humans, and defines the ways in which technology may be applied. Moreover, the definition of the organization, with formal and informal networks of command, will determine the rules for information flow throughout the organization. Organizational environmental factors change over time, as regulatory changes, disruptive technologies and other external factors affect them. However, it is always through these organizational aspects that the cues and responses to and from the man-machine systems environment are filtered.

In their complex systems research, Bartolomei et al. (2012), describe a model of systems that included five major domains: The social domain (of team and enterprise behaviors), the Functional and Process domains (of system functions and human tasks), the Technical domain (system components and interfaces) and the environmental domain of physical, economic and technological pressures. This model migrates over time as the pressures accumulate and the system adapts.

Mapping between the functional and process domains describes the traditional human factors engineering method mapping of missions (goals) to functions and tasks (Sanders & McCormick, 1993). It is here where methods focusing on the HMI provide the most utility. However, for socio-technical systems where many of the factors affecting system behavior do not have an analog on the HMI, the concept of JCS becomes much more powerful. Figure 2 depicts this model.

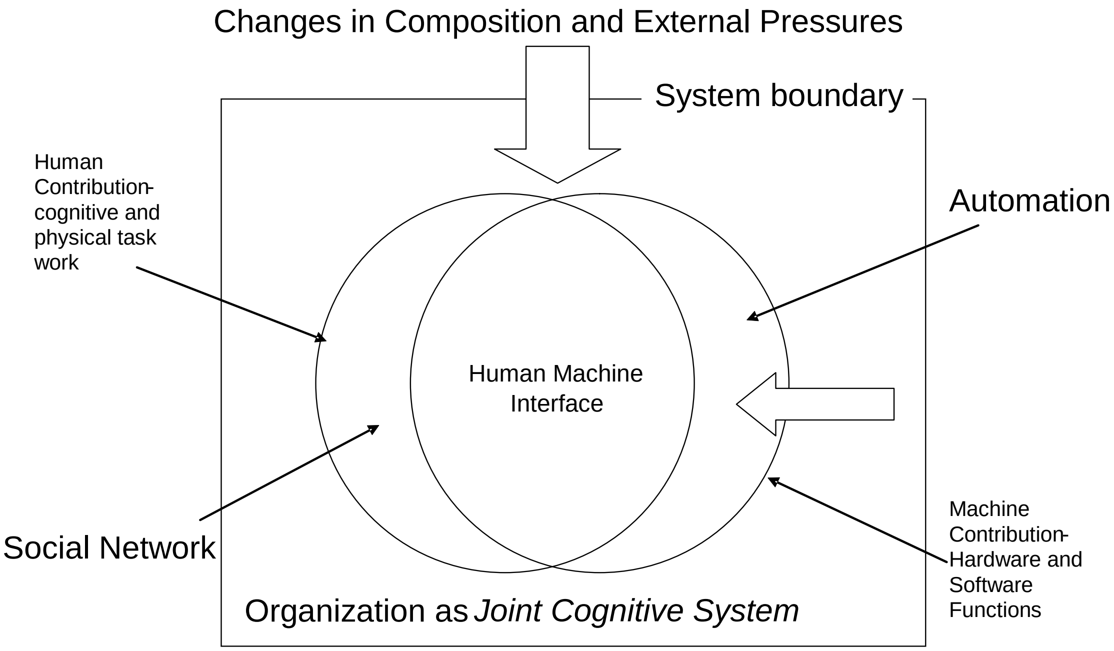

Figure 2 . Components of the Joint Cognitive System. This figure depicts the system and its constituent components from both the Psychological and Technical perspectives. Adapted from (Bartolemei, Hastings, De Neufille & Rhodes, 2012).

The previous sections discussed complexity as it relates to systems design. It introduced some research communities and concepts that arose to address this complexity. The next section will describe the practical methods which have been developed to apply this new science. It starts with a discussion of cross-functional teams as a management tool used to develop systems.

Cross Functional Teams in Systems Engineering

Human Factors Engineers have been participating in system design for decades. Initially, they were brought in after-the-fact, and asked to address problems of usability that resulted in undesirable performance (Sanders & McCormick, 1993). Recently, with the adoption of Human Systems Integration (HSI), especially within the aerospace and defense industry, they are available much earlier in the design process to provide review and input. Additionally, the 1980s different disciplines have participated in cross functional teams called *Integrated Product Teams (IPT*), where all stakeholders, including human factors, could be represented as decisions were made (Kossiakof & Sweet, 2003). The power of the IPT model is in having the stakeholders present, but it does not, in and of itself, provide any tools to enhance communications among the stakeholders.

While HSI presents the methods and tools of human factors (and the other HSI domains) to the systems engineer, it has not yet provided robust guidance in truly integrating the domains, other than to clarify that it is important to focus on requirements (Pew & Mavor, 2007), as this is the tool used by systems engineering to affect the outcome of the final system.

However, comparing the systems engineering method (Kossiakof & Sweet, 2003) and the human engineering methods (Sanders and McCormick, 1993), as they pertain to functional analysis and allocation, it is apparent that something is different. Since systems engineering focuses on allocation of automation to hardware and software components, while human engineering addresses automation explicitly as an allocation of functions to hardware, software and human components, clearly something is qualitatively different in their understanding of *function*. In the systems case, function is something that exists independent of time (Kossiakof & Sweet, 2003), however, in the human engineering case, automating tasks exists conterminously with functional allocation, and tasks have an implicit characteristic of time, as they can be said to have duration (i.e. a start and stop time).

The preceding sections presented a review of the characteristics of complex systems. It was followed with discussion of the various methods that have evolved to address complexity in design, especially emanating from the human factors community. The next section will provide a discussion of the mental models necessary in individuals and teams to communicate on teams employing these methods, as well as describe some graphical methods for conducting this communication.

Mental Models and Graphical Models

Individual Mental Models

Mental Models evolved as a concept to explain how people created structures in long term memory to arrange acquired information that would later be used to make decisions regarding interaction with the world. The term *mental model* was first used in 1943 by Craike, but gained notoriety in the 1970's (Johnson-Laird, 1983). Since then, considerable research has been devoted to mental models to address deficiencies in the stimulus-response model dominant in psychology up to that time (Pew & Mavor, 1999). Mental Model theory is useful to and has contributors from a number of different domains, from psychology to linguistics. Jens Rasmussen (Rasmussen, 1990) provides a convenient human factors definition of mental models, as "A mental model of a physical environment is a causal model structured in terms of objects with familiar functional properties."

Mental models cannot be observed directly; therefore much disagreement exists as to their nature and composition. The dominant theory is that they are “iconic” composed of visual images. A contravening theory proposed by Pylyshyn states that images are only related in a *phenomenological* way and rely on a superficial experience to generate a fairly hollow physical representation for association. Thus the real mental model is composed of *propositional computations*, similar to formal logic, such as A is greater than B, C is greater than A, thus C is greater than B. (Johnson-Laird, 2006). In both cases, they identify the understanding relationships and connectedness between concepts as essential to the model (Mackiewitz & Johnson-Laird, 2011). Similarly, both posit that this understanding is used to create a “mental simulation” of expected causality. Prediction of future states or future desired states is a function of perceiving the current state of the world, orienting these perceptions to the context of the understood problem domain, and selecting an expectation of action. Observation of actual future states will cause the understanding of context to evolve over time (Neumann, Badke-Schaub, & Lauche, 2006).

Technical Team Mental Models

The concept of a *team mental model* was proposed by Cooke, Salas and Cannon-Bowers in the 1990’s (Neumann, Badke-Schaub, & Lauche, 2006). Team mental models extend the idea of shared cognition to posit that a working team of individuals must have a joint vision of the task to be performed and the nature of the team, and their place in it, in order to execute the task. A critical premise of this research is that engineering teams, like any other team, rely on shared mental models to conduct their work. This premise is well supported by research (Avnet, 2009; Lim & Klein, 2006). As described earlier, changes in the systems engineering community acknowledge the need to include additional stakeholders in systems of increasing complexity, such as introduction of human systems integration (HSI) as a systems engineering activity within the aerospace and defense industry. HSI is defined as a specialty engineering community including the traditional social domains of human factors, training specialists, environmental safety and occupational health (Mueller, 2008). At the same time, considerable research has been devoted to understanding team dynamics and the factors that enable such diverse teams to operate in a high performing way (Neumann, Badke-Schaub, & Lauche, 2006; Cannon-Bowers & Salas, 1998; Avnet, 2009; Defranco, Neil, & Clariana, 2011; Lim & Klein, 2006). Recent research has suggested that the similarity between team mental models is actually more important than the accuracy of the team mental model (Lim & Klein, 2006). Early work focused on determining the factors affecting team high performing teams in tactical tasks (Cannon-Bowers & Salas, 1998); later work has expanded this to identifying the factors in developing high performing design teams (Avnet, 2009; Defranco et al., 2011; Lim & Klein, 2006; Neumann et al., 2006). A common theme throughout all this research is that team members of high performing teams had a shared vision of the current and future state of the tasks ahead of them, and they continuously updated this vision through communication and coordination (Defranco et al., 2011).

> Premise #1 – Success performance of human factors and technical practitioners working in concert on a system design team require congruence in their shared mental model of the state and purpose of the design.

As discussed in the section on individual mental models, mental models include physical and behavioral representations of the world, used to conduct mental simulations to determine future states. This representation of the physical and functional relationships of the world, as well as their current states, must be continuously updated, and synchronized, between members of a team in order to establish team situation awareness (Cannon-Bowers & Salas, 1998). For activities where the goal is to describe the functional architecture of something that the team is to build (e.g. a UAS), it could be argued that the models they develop actually are equivalent to the mental model they hold of the system (at least once the final model has been agreed upon by all stakeholders). As stated previously, there is some disagreement in the nature of mental models; however both camps generally agree that relationships are critical. To preserve that characteristic, and to support the logical argument of equivalency, this research will focus on relationships and reduce the dependency on experience for iconic representation by both using students who lack direct experience and by providing the structural representation of the problem domain (i.e. context diagrams as depicted in the appendices). Thus, for the purposes of this research, two problem statements are provided that describe the relationships between elements, in the closed case explicitly as physical interfaces, and in the open cases as information exchanges. Furthermore, functional flow represents relationships as casual propositions (i.e. how the system reacts over time) and are therefore consistent with this aspect of the task representation of a team mental model.

> Premise #2 – Descriptions of system functionality represent the behavioral understanding of the modeler with respect to the problem domain, and can thus be seen as equivalent to the domain component of the modeler’s mental model.

Measurement of mental models has evolved from questionnaires and surveys to evaluation of concept graphical models developed by experimental participants. A particularly powerful way to execute this technique (Lim & Klein, 2006; Defranco et al., 2011) used concept maps and a tool called “pathfinder” to create average hierarchies from the highest weighted paths and then measure individual concept graphs against the average. A graph with a single *spanning tree* (Buede, 2000) of nodes arranged into horizontal levels is by definition a *hierarchy*, and performing this activity on a functional network provide the functional decomposition used by technical and social domain practitioners in their modeling. Lim used this technique to demonstrate the impact of accuracy and similarity on team performance, with performance evaluated qualitatively, and mental models measured quantitatively. Defranco et al. (2011), used the concept map to measure the mental models of his participants and determine if his proposed “Cognitive Collaborative Model” could help them improve their joint understanding of a design problem and reduce the variation between the team member mental models. This research proposes using the concept map technique to measure mental models and determine the effect that complexity and training or experience has on that variation. When the team task is the development of a system, the graphical models used to describe aspects of the system may in fact equate to mental models, at least the portion associated with the representation of the structure and behavior of the system.

> Premise #3 – Graphical models of the functional architecture of a system can be used to measure the domain component of the mental model of the designer. They describe the goals/means decomposition of the system through functional relationships.

Designers, both engineers and human factors, commonly use a variety of graphical model formats and methods throughout the system development process. This next section will describe some of these.

Graphical Models Used in the Technical Domains

The Object Management Group (OMG), formed in the 1990s by three leaders in software engineering at IBM, proposed the Unified Modeling Language (UML). The UML was intended originally as an aid for software developers to exchange design information, and in the ensuing years, it has matured significantly. It has been adapted into the Systems Modeling Language (SYSML) as well, to capture the design elements of not only software, but hardware, humans and other elements of the system. The use of UML and SYSML provides a semantically precise mechanism for capturing the elements of design: utility, structure and behavior, as well as the nature of the relationships between these elements (Friedenthal, Moore & Steiner, 2008). These languages have evolved sufficiently to enable systems design through a model based systems engineering (MBSE) process, where the subjective interpretation of the natural language in prose specifications can be replaced with fixed relationships, semantically precise graphical model elements and a system of logical tools for validating the completeness and consistency of these models. It is because of this richness that this research assumes that validated UML/SYSML models will measure the mental models of design practitioners working on a system.

Systems engineering and software engineering originally relied on a *structured method*, whereupon the system functions were first decomposed and then allocated to components. Structured methods would create a hierarchical breakdown of system functions, which would then be allocated to either hardware or software, and then to specific components (Kossiakoff & Sweet, 2003). Figure 3 depicts the functional block diagram (FBD) hierarchy used in structured methods.

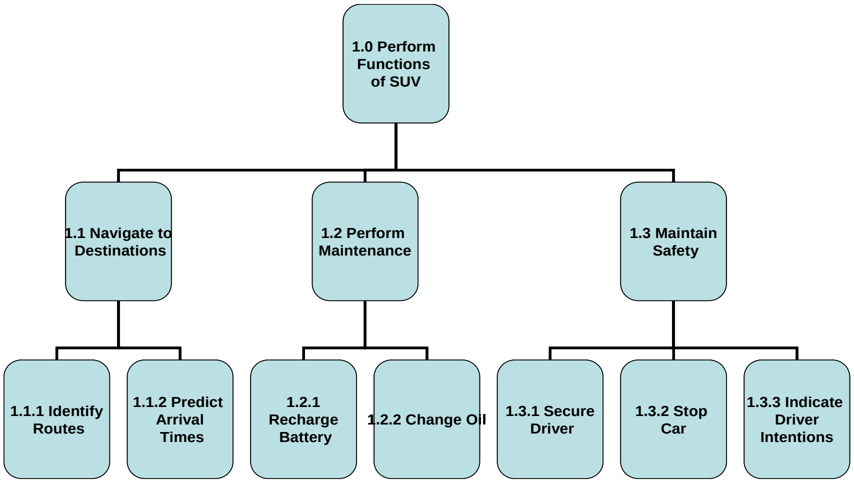

Figure 3. Function Block Diagram. This figure illustrates the hierarchical nature of Functional Decomposition as applied to a plug in hybrid SUV.

Hierarchical decomposition of functions can also be arranged to capture aspects of time. These are known as *functional flow diagrams.* Adding in signaling and data flow makes for enhanced functional flow block diagrams" (EFFBD). EFFBD include looping, logical AND and OR features, and branching/selection logic. As with FBD, they can be decomposed to increasingly lower levels of detail.

Recent methods developed for systems engineering rely on object oriented (OO) methods. Object orientation arose from the software engineering community about 15 years ago (Friedenthal et al., 2008). Both object-oriented and structured methods rely on the notion of function as a core construct defining system behavior. Within UML and SYSML, the modeling language used for object oriented methods, functions are defined within activity diagrams. UML version 2.1 extended the earlier UML activity diagram to include all of the information contained within EFFBD used by the earlier structured approaches (Friedenthal et al., 2008). The EFFBD convey more information than the functional hierarchy by displaying functions and data flows between functions, as well as details of that flow, whether it is synchronous, continuous, or asynchronous as well as its content or medium (information, mass or energy, etc.). Figure 4 provides an example EFFBD in classical structured notation.

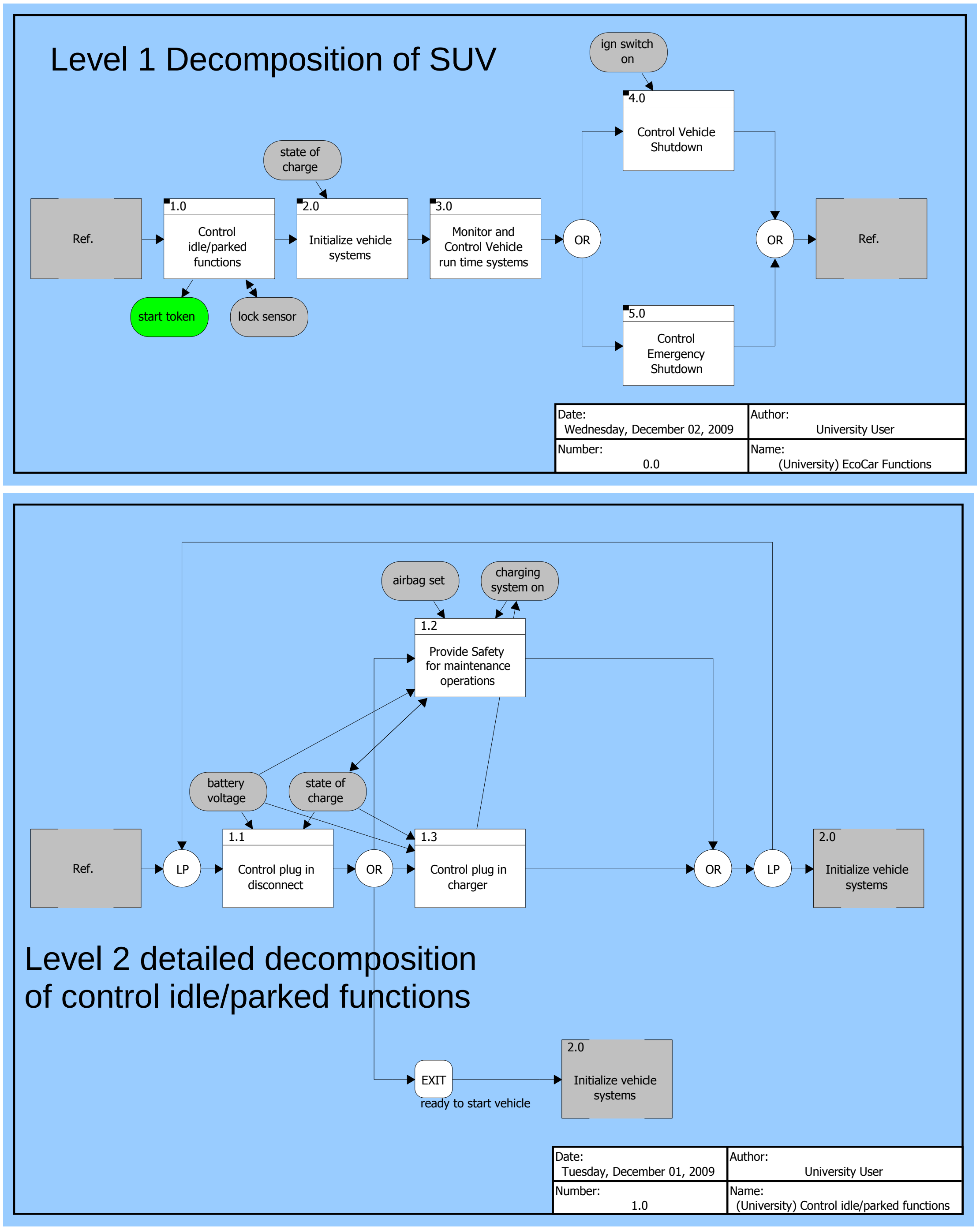

Figure 4. Example Enhanced Functional Flow Block Diagram (EFFBD) drawn in Vitech CORE™. This figure illustrates decomposition of functions at multiple levels of detail for a plug-in hybrid sport utility vehicle (SUV). Circles represent and for parallel, OR for optional and LP for looping functions. Functions are represented as white boxes. Grey boxes link related functions in other diagrams. Lozenges represent data, which may be inputs or outputs of functions as indicated by arrows.

Graphical Models Used in the Social Domains

Practitioners in the social domains also use graphical models. A common example is the task analysis representation. Annet (cited in Hollnagel, 2003) describes an established task analysis method, the *hierarchical task analysis* (HTA). This method was initially developed in the 1960s as an alternative to time and motion studies. It bears some similarity to the structured analysis methods of classical systems engineering, as it decomposes problems from high levels to increasingly granular levels of detail, in assigning the relationship between form and function, and describing functions in terms of inputs, outputs, and processes.

The HTA has undergone a steady evolution, through the work domain analysis (Vicente, 1999), to the *functional abstraction hierarchy* (Hollnagel, 2003) and many variations in between. Each of them presents an *abstraction-decomposition* space (Vicente, 1999) that moves from lesser to greater levels of specificity and concreteness, starting with abstract statements and ending with actual systems in operation providing value. These models start with a definition of *goals* (as attributes of the users and not having meaning without them) and decompose to the *means* to accomplish these goals. As the models become increasingly fine grained, they move from statements of purpose, to described behaviors to design specifications. Woods & Hollnagel describe this as answering “why, what and how” with each successive level providing the components which address one of the previous levels means as its own goal as in Figure 5.

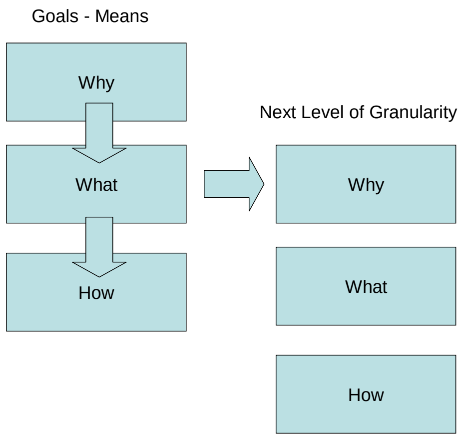

Figure 5. Goals Means Decompositions. Adapted from Woods & Hollnagel, 2006. Each level represents the association of why something is done, what is done and how it is done. Each level of "what is done" is the *why* for the next level of detail.

Other researchers have different perspectives, for example Lintern reinterprets it as a "what, what, what" set of questions, and makes the argument that the concepts should stay behavioral in scope (Lintern, 2009). He warns against making structural assumptions too early. Despite differences, however, both authors indicate that the lines of demarcation can be somewhat arbitrary, and state that the characteristics for splitting up concepts between parent-child and peer-to-peer can change as the hierarchy progresses.

An example of graphical languages with formal semantics used in human factors research is *Petri Nets* (short for Place/Transition Network). Petri Nets are a graphical language presented as a formal predicate calculus (Esparza & Lakos, 2002). There has been research into the human factors community to translate Petri Nets into formal language for human factors data (Jonassen et al., 1999), and the principal investigator has developed several methods and tools based on Petri Net research, and applied them on a half dozen programs within the aerospace and defense industry to capture and evaluate human factors data (Gordon, Burns, & Giebenrath, 2005). Additionally, the authors of the *Applied Cognitive Work Analysis* process described in the cognitive systems engineering section developed a graphical notation for capturing goals, processes and decisions within functional abstraction networks (FAN). Figure 6 shows an example FAN.

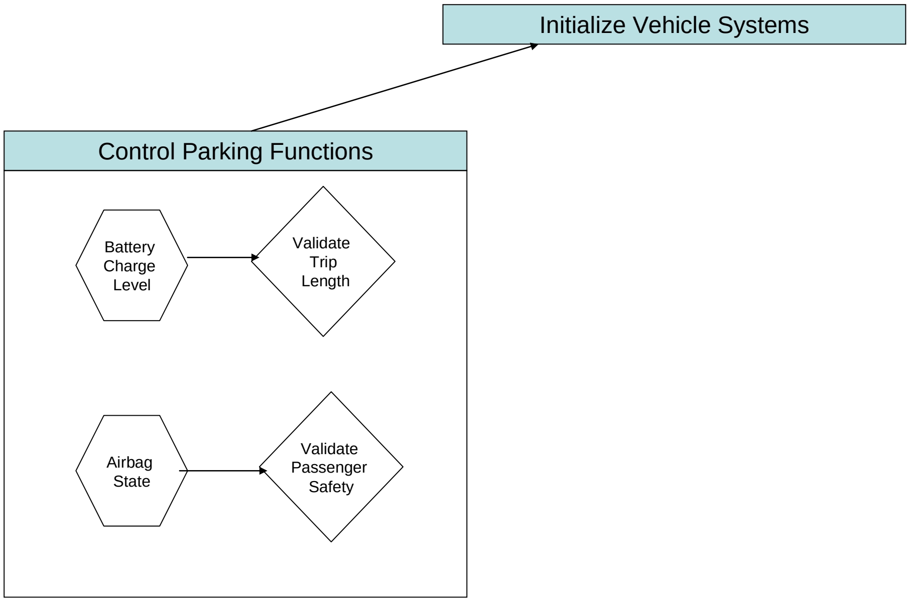

Figure 6. Functional Abstraction Network (FAN) of the Plug-in Hybrid SUV of Figure 4. Adapted from Potter, Gualtieri & Elm, (2002). Goals are listed at the top, decomposing to related functions, which are supported with sub functions and information concepts from the work domain, represented as diamonds and hexagons respectively.

The similarity between graphical models used to capture functional flow or activity modeling in the technical domains and the goals/means decompositions of the social domains is critical to this research. A review of the literature shows, at least in principle, both technical and social practitioners must think about the behavioral relationships of the system that they believe they are building. It also demonstrates that technical and human factors practitioners have developed graphical methods to aid in sharing this conception of the problem, and that a shared conception is essential to team performance. The challenge is then for systems where the boundaries that define what is, and isn’t, a goal of the system, and how those goals relate back to the structures used by people to achieve them is increasingly complex. In these systems, the goals and functions are less easy to describe with team consensus, graphically or otherwise.

Problem Statement

The purpose of this research was to assess the degree of shared understanding among technical (systems engineering) individuals and social (human factors engineering) practitioners as they attack the development of systems by developing functional models. Each functional model is composed of "concepts" (i.e. functions and sub-functions) identified by the participants and arranged hierarchically in two levels of decomposition under high level organizing concepts provided in the problem statements. Additionally, this study also investigates whether the complexity of the system under development has an effect on the level of mutual understanding among the disciplines.

Hypotheses

Two basic research questions are addressed in this experiment. The first is whether the degree of mental model agreement among individuals addressing the same problem is affected by whether the individual mental models are compared against a mental model developed with practitioners from the same educational/practice specialty (systems or human factors) or when participants are compared against a model developed from the opposing educational/practice specialty. This question is addressed by comparing the individual participant model to a weighted "average" model defined within their practitioner group (In-group), and then again when comparing against the average models for the opposite group (Cross group). This researcher expects that the mean Cross-group model agreement score of participants shall be smaller ($\mu_{\text{Crossgroup}}$) than the In-group score (μ_{Ingroup}).

The second question is whether the level of agreement for participant models is also affected by the complexity of the system under consideration. This question is investigated by comparing the participant models measured against average models for the two problem types: simple/closed and complex/open. It is expected that the higher level of problem complexity (i.e., “open”) will yield lower average agreement scores when compared to the easier problem (i.e., “closed”). It is also expected that a two-way interaction effect will appear, with the difference between Cross-group and In-group scores compared by group greater in the open case, than in the closed case. Figure 7 summarizes these questions and associated formal hypotheses.

The researcher expects that the means of the semantic agreement scores in the closed level case will be very similar, as the functions required are defined largely in the provided problem statements, although H_{1c} may still be valid because of semantic differences in the open case. Figure 8 illustrates these expectations.

Q1 = Is the level of agreement between participant models affected by whether they are compared against a weighted average domain model calculated from their own group (In-group) or from the opposing group (Cross-group)?

H1 The level of agreement in the Cross-group scores will be lower than the In-group scores for all three calculations of agreement (Horizontal, Vertical Semantic)

- H1a = $\mu_{InGroupHorizontal} > \mu_{CrossGroupHorizontal}$
- H1b = $\mu_{InGroupVertical} > \mu_{CrossGroupVertical}$
- H1c = $\mu_{InGroupSemantic} > \mu_{CrossGroupSemantic}$

Q2 = Is the level agreement between participant models affected by the complexity of the system under consideration?

H2 = The level of agreement in models scored at the simple/closed level will be greater than the open/complex case for all three calculation of agreement (Horizontal, Vertical, Semantic)

- H2a = $\mu_{HorizontalClosed} > \mu_{HorizontalOpen}$
- H2b = $\mu_{VerticalClosed} > \mu_{VerticalOpen}$
- H2c = $\mu_{SemanticClosed} > \mu_{SemanticOpen}$

H3 = There will be a two-way interaction effect, where the Cross-group and In-group differences are larger in the complex case than in the simple case for all three calculations of agreement (Horizontal, Vertical, Semantic)

- H3a = $\mu InGroupHorizontalClosed - \mu CrossGroupHorizontalClosed > \mu InGroupHorizontalOpen - \mu CrossGroupHorizontalOpen$
- H3b = $\mu_{InGroupVerticalClosed} - \mu_{CrossGroupVerticalClosed} > \mu_{InGroupVerticalOpen} - \mu_{CrossGroupVerticalOpen}$
- H3c = $\mu_{InGroupSemanticClosed} - \mu_{CrossGroupSemanticClosed} > \mu_{InGroupSemanticOpen} - \mu_{CrossGroupSemanticOpen}$

Figure 7. Hypotheses evaluated in the experiment. Means are representative of variable combinations which are summarized in table 4 in the Method Section. Means (μ) for semantic, horizontal and vertical agreement according to the test conditions manipulated as independent variables – In-group or cross group model domain comparison and open or closed problem type. All hypotheses are evaluated using the interaction of the group variable instead of main effects to control for group differences not related to just being in the HF or SY sample.

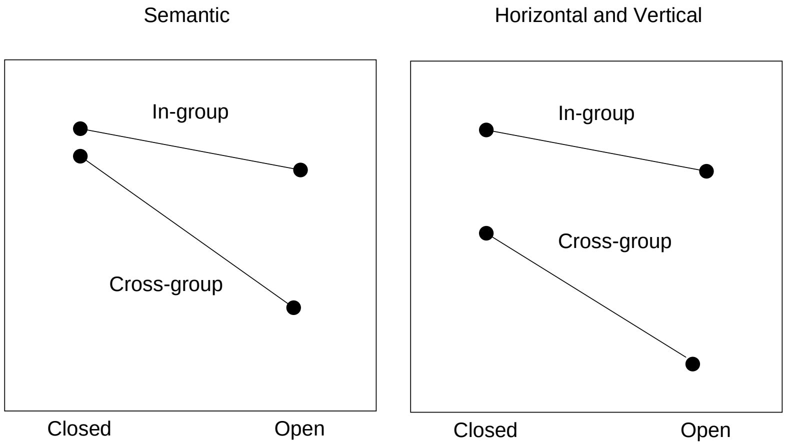

Figure 8. Graphical depictions of expected values in the Experiment for Semantic, Horizontal and Vertical Agreement by Model domain and Problem Type. The expected 2 way interaction between IV1 (in-group/cross-group) and IV2 (closed/open) is apparent in the non-parallel lines.

Design

The experiment is a modified two by two by two design. The first Independent Variable (IV) is the education/practice specialty (HF or SY) to which the participant belongs, by virtue of education and experience. The design is completely within for both the second IV; comparison model domain type (i.e. whether agreement scores are calculated against In-group or Cross-group average models), and the third IV, problem type expressed as level of complexity (closed or open). The Dependent Variables (DV) are the weighted scores of semantic, vertical and horizontal agreement (defined in the method section) of the participant model to the respective group model.

Independent Variables

The first Independent Variable (IV) is education/practice specialty type. The experiment selected participants from two specialties, social (human factors) practitioners and technical (systems engineering) practitioners. Specialty type is an inherent characteristic of the participant, based on this experience and education.

The second IV is comparison model domain type. Model domain type describes the relationship between the specialty type of a participant and the specialty type used to define the "average" model to compare against the participant model. In-group model domain means the participant and model are the same type (HF compared to HF; SY compared to SY), and Cross-group model domain means they are of opposing types (HF compared to SY; SY compared to HF). "Average" models are defined as the collection of weighted contribution scores for all concepts defined by participants of a particular specialty (i.e. the more instances of a participant identifying a concept, the greater its score contribution to the "average"). The researcher calculated In-group models, one for systems engineers and another for human factors engineers. In-group models were calculated by removing an individual's contribution to their In-group average (subtracting one from the number of instances a concept is identified in the weighting calculation). Cross-group models were calculated from the same raw data, but were scored differently (including all instances of a concept), as the participant scored Cross-group was not contributing to the "average" the same way an In-group participant is. Each participant is then compared against the "In-group" model average to calculate In-group agreement score. A second score for Cross-group was calculated for each participant by comparing his or her model against the average model calculated from the opposite specialty.

The third IV is problem complexity. Study participants completed functional analyses of two scenarios or "problems." This experiment used two levels of problem complexity. The first level was a *closed system*. The closed system scenario does include automation, however it is closed because it also includes a high degree of operator monitoring and control, fixed system boundaries, and well-defined interfaces. The operational life cycle of the closed system is also well defined. It is a “simple” problem, and it lends itself to top-down decomposition from an overall goal and phase model. The second level is for the *open system*. This open system scenario has ill-defined boundaries, an undefined requirement to transfer some functionality to and from external interfaces over which the designer has no direct control, and would reconfigure itself, and define its purpose, dynamically while in use. These characteristics indicate that the open system is “complex.”

Dependent Variables

The Dependent Variables (DV) are related to the agreement between participant models and model domain type (In-group and Cross-group) models. This study used three measurements of agreement:

- Semantic Agreement – the same child node sub-functions/means are defined. Child nodes represent concepts at the lowest level of decomposition/aggregation within the functional hierarchy.
- Vertical Agreement – The same parent node goals/purposes/functions are defined. Parent nodes include the concepts between the provided phase nodes and the leaf nodes of a given hierarchy.
- Horizontal Agreement – The same parent/child relationships for goal/functions and phases are defined. The association of each leaf node to the parent node it branches defines parent-Child relationships.

This design yields twenty-one possible combinations of IV/DV scores. Hypotheses are mapped to nine, including main effects of model domain and problem type, and the two-way interaction of model domain and problem type. Any significant results are to be compared against the interaction of that case with educational specialty to control for other potential confounds. Table 4 lists these effects.

Table 4
Potential observable effects in this experimental design

IV Comparison	Potential effects in DV		
	Semantic	Horizontal	Vertical
Specialty (HF vs. SY)	main effect on semantic agreement	main effect on horizontal agreement	main effect on vertical agreement
Model Domain (In vs. Cross)	main effect on semantic agreement	main effect on horizontal agreement	main effect on vertical agreement
Problem Type (Open Vs. Closed)	main effect on semantic agreement	main effect on horizontal agreement	main effect on vertical agreement
Model domain * Problem Type	interaction effect on semantic agreement	interaction effect on horizontal agreement	interaction effect on vertical agreement
Specialty * Model domain	interaction effect on semantic agreement	interaction effect on horizontal agreement	interaction effect on vertical agreement
Specialty * Problem Type	interaction effect on semantic agreement	interaction effect on horizontal agreement	interaction effect on vertical agreement
Specialty * Problem Type * Model domain	interaction effect on semantic agreement	interaction effect on horizontal agreement	interaction effect on vertical agreement

Note: Hypotheses are associated with the potential effects highlighted in grey. To present valid results, the associated interaction effect with specialty and the comparison representing a hypothesis should not be significant, as this would indicate potential confounds from the specialty dynamic not associated with training and experience.

Participants

The experiment participants were current Human Factors and Systems Engineering graduate students pursuing master's degrees at a southeastern university. Thirty-five students provided data, nine HF students and 26 SY students. Seven HF and 12 SY students' data were usable. Reasons for rejection of data are detailed in the Results section. The usable data sets from human factors included five female and two males, from systems engineering nine males and three females. No evidence suggests that gender would have an effect on their ability to perform this task, and this distribution reflects the student populations for these degree types as a whole. Additionally, submittal of the post-exercise questionnaire forms was not complete by all participants, although presence of the form was not included in evaluation criteria for acceptance or rejection of the data sets provided.

Participants were categorized into two specialties of practitioners, systems engineers (representing the technical engineering domain) and human factors engineers (representing the social domain), at the novice/apprentice level. Participant students had classes or work experience that provided exposure to functional analysis methods for their respective specialty. None of the participants claimed to possess any “Cross-group” experience.

Materials

The following tools and materials were used to conduct the experiment, and are summarized in subsequent subsections:

- Windows computer terminal and internet (for XMind download) for each participant (working files may be stored on personal network drives)
- XMind Software
- Consent Form (Appendix A)
- General and XMind tool Instructions (Appendix B)
- Demographics questionnaires (Appendix C)
- Post experiment questionnaire (Appendix D)
- Closed system problem statement (Appendix E)
- Open system problem statement (Appendix F)
- Writing implement (i.e. pen/pencil)
- Scratch pads were optional and could be used, but weren’t evaluated

  - MSOffice (the researcher used both MSExcel and MSAccess for data reduction tasks)
  - IBM Statistical Package for Social Sciences (SPSS 19)

Apparatus

XMind software tool. XMInd is a hierarchical brainstorming tool that uses a graphical user interface to allow users to decompose and link concepts in a visual "wheel style" fashion. XMind is a very simple to use tool, and only a few of its features are required to perform the tasks within this experiment. General instructions on use of the tool, as well as specific instructions on formulating and rendering functional models within XMind are presented at the beginning of the data collection. Written instructions are presented as Appendix B**.**

Demographics data collection. The demographics questionnaire was used to collect data such as gender, age, and Cross-group experience via industry or education. The form screen participants for primary language to control for non-English native speakers- as this might impact the ability of the participant to frame and describe the problem in the same way as the other participants. The questionnaire is presented as Appendix C.

Post experiment questionnaire. The post experiment questionnaire asked the participants to define a series of terms, such as "function" and "task" that are used within the modeling activity. The Post Experiment Questionnaire is presented as Appendix D.

Design Problem Scenarios. The two scenarios are selected from the Unmanned Aircraft Systems (UAS) problem domain. The "simple" or "closed" problem was the first scenario. It described a software automation upgrade for the mission segment of ground control software. The problem statement defined customer capabilities, system boundaries, system life cycle, and system interfaces. The second scenario, the "complex" or "open" problem, described a deployed mission planning system for the same UAS, which would hypothetically be installed in a number of different vehicles and provide levels of automated support to a number of potential external interfaces. The customer wished to accomplish a personnel reduction, but no guidance was given for the scope of automation to achieve it. The problem statement did not define boundaries or the exact format of interfaces, and the system description implies that it was to be deployed and used in unforeseen ways. Details of the Scenarios are provided in Appendices E (closed) and F (open).

Procedure

The researcher attended one class each for the systems engineering and human factors engineering students. The experiment began with the researcher explaining the informed consent and risks to the potential participants. The researcher explained the problem statements, the use of XMInd, the constraints on the participants, and the objective of the research. Each participant received a numbered packet containing the demographics questionnaire, the two problem statements representing the system levels and the post experiment questionnaire.

Immediately after handing out the materials, students returned the signed forms. The researcher conducted an hour-long guest lecture on the relationship between use cases, functions, and requirements. This was to help synchronize and control for the “task based” component of the students’ mental models as the Defranco et al. (2011) experiment illustrated the impact of task synchronization.

Following the lecture, the researcher provided participants with general instructions in using XMind, as well as detailed instructions on developing functional hierarchies within XMind. These instructions included use of “present tense verb-object” protocols along with some examples to reduce semantic noise. The participants create the functional hierarchy visually within the XMind tool and then saved it as a word document, a web page, or an XMIND file. The hierarchy started with the first two levels provided in the problem statement (mission/phase) and asked the student to identify the next two levels (function/sub function) as depicted in Figure 9.

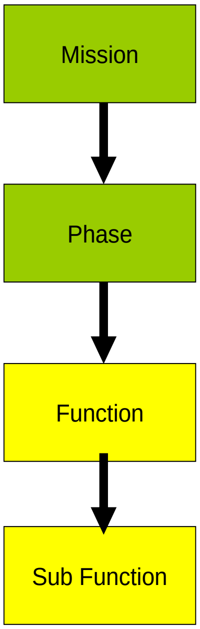

Figure 9. Hierarchy of Concepts in Functional Decomposition. Objects in Green were provided as part of the problem statement, objects in yellow were to be defined by the participants.

Odd numbered students performed the closed problem first and open problem second and even numbered students performed the open problem first and closed problem second to counter balance and mitigate learning effects. Participants performed work outside of class on their own time and in their own chosen locations. The researcher answered questions via e-mail regarding use of the XMind tool. No students asked questions regarding UAS subject matter expertise, although the experimental protocols allowed it. The exercise was not timed, but the researcher informed students that it should take less than an hour for each. Informal feedback to the researcher from some participants indicated that problems took between 30 minutes and 3 hours to complete.

Students performed the example problems as homework assignments for one class and as extra credit for the other class. Participation in the experiment was not required for the grade. No students elected not to participate, however, if they had, they would have been able to complete the assignment for credit independent of the experiment.

Participants were asked to provide written definitions of the listed terms in the post experiment questionnaire, as they understood them. After the students completed their models and the demographics and post experiment questionnaire, they emailed their results to the research at the e-mail provided on the consent form. Students performed the example problems as homework assignments for one class and as extra credit for the other class. The grade did not require participation in the experiment. No students elected not to participate, however, if they had, they would have been able to complete the assignment for credit independent of the experiment. Table 5 summarizes the experimental method.

Table 5.

Summary of steps in performing the experiment.

Lead	Step	Description
Researcher	1	Describe Experiment and Risk/Benefit
Researcher	2	Obtain Signed Consent Forms
Researcher	3	Conduct Lecture on Functions and Requirements
Researcher	4	Describe XMIND tool
Researcher	5	Describe Problem Statements
Researcher	6	Describe Experimental Protocols
Researcher	6.1	Proper construction of function and sub-function concepts
Researcher	6.2	Problem Ordering for counter balancing
Researcher	6.3	Submission of Materials
Researcher	6.4	Time and reference constraints
Participants	7	Develop Experimental Data
Participants	7.1	Download and install XMIND tool
Participants	7.2	Read Problem Statement
Participants	7.3	Perform First Problem
Participants	7.4	Perform Second Problem
Participants	7.5	Save Model to File
Participants	7.6	Complete demographics survey
Participants	7.7	Complete Post experiment questionnaire
Participants	7.8	E-mail Results to Researcher

Results

Data Reduction

Appendices G (parent concepts) and H (child concepts) present the raw data. After collecting all of the participant models, the researcher evaluated the submitted data sets for completeness and compliance with the parameters of the experiment. Inclusion in the data analysis required the participant to have provided two levels of properly constructed functional statements representing goals/means decomposition below the provided mission phases for both the closed and open problem cases. Thirty-five students ultimately provided data, nine HF students and 26 SY students. The researcher obtained usable data from seven HF and 12 SY participants. The remaining data were rejected for the following reasons: the submitted model represented functional allocation instead of functional decomposition (five participants), the model did not properly use the provided mission phases (one participant), only one problem type was submitted (six participants), the model described requirements instead of functions (one participant) or the model described physical rather than functional hierarchies (three participants).

The next step in data reduction was a cleanup of the concepts in each model to eliminate non-semantically relevant differences in the literal text. The experiment instructions asked participants to use a strict present tense verb – direct object noun format for describing functions to help reduce noise due to unimportant differences. Not every participant complied, so any non present-tense conjugations, subjects, adverbs, or extraneous clauses were also removed from their model concepts. Non-substantive differences like “transponder” vs. “IFF” and “radio” vs. “transmitter” were removed; however differences such as “communicate” vs. “interface”, “flight” and “aircraft”, and “radio” vs. “data link” which implied true semantic difference were left in the data. Lastly, some participants did not define subordinate functions with verbs, but it was clear these subordinate functions represented more detailed sub functions of their parent, rather than allocations. In these cases, the parent "verb" was used to define the children concepts. In one example, "control systems" was followed by a list of systems, such as parachute, payload, etc. It is obvious that "control systems" was not allocated to them, but that those children represented details describing the systems to be controlled.

The original thesis proposal described a process for determining an "average" model using the concepts identified by at least 32% of the participants within a specialty (i.e. ~1-2σ). This method was selected because randomized trials conducted to validate the experimental design showed that this method had a lower signal to noise ratio. Upon initial analysis of data during the experiment, the total number of concepts achieving the 32% (1-2σ) frequency rate was very small in comparison to the overall list (less than 25%). In order to avoid floor effects (i.e. setting a low bar for agreement) the researcher used a weighted average scheme instead, where all concepts defined by a specialty would be used, and their relative weighted contributions summed. This technique thus included all concepts identified within one specialty type, opening the aperture of analysis and lowering the signal to noise ratio. The researcher chose this as a more conservative scoring mechanism, providing less opportunity for Type 1 error, as depicted in Figure 10.

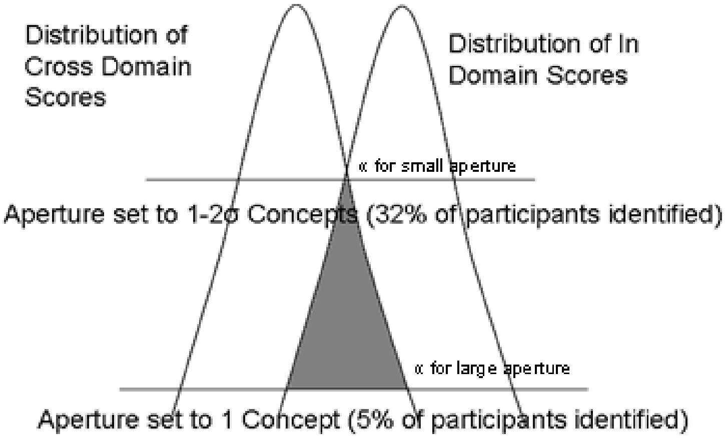

Figure 10 Effect on notional score distributions based on analysis aperture setting. Original proposal required 32% of participants to identify a concept for it to be included in the average. Actual experiment relied on a single participant to identify the concept, but concepts were weighted based on frequency of occurrence in models.

Sub function concepts identified by each participant were collected in one MSExcel™ spreadsheet for use in semantic analysis. Function concepts and associated phases for each participant were listed in a separate worksheet for use in horizontal and vertical analysis. Each concept was listed with the participant that generated it, the Specialty (HF or SY) for which the participant is associated, and the Problem Type (closed or open) for which the concept was identified. This entire table was input to MSAccess™, which was the tool used for data collation and derivation of agreement values.

As expected, the frequency of participants identifying a particular concept in their individual model (such as "Maintain Communications") obeyed a Pareto distribution. To create the agreement scores for each concept's contribution, a normalization routine was developed. For each combination of educational specialty and problem type (i.e. HF open, HF closed, and SY open, SY closed), the following procedure was used:

1. The child list was copied to another worksheet and all duplicate values removed.
2. This list was then used with Excel COUNTIF function to count instances of all unique concepts in the child list.

3. These instances were divided by the total number of concepts on the child list to provide a weighted contribution for that concept for Cross-group comparison.
4. The number of instances was reduced by one (to remove the contribution of an individual from the average) and step 3 was repeated with the new total to find contributions for In-group comparison.
5. These were input to MSAccess to provide the scoring contribution tables for each analysis.

An example of the MSAccess input table is presented in Table 6.

Table 6
Example of Concept Score Contributions

Model Concept	In-group score contribution	Cross Group score contribution
transfer control	0.07	0.02
transfer payload	0	0.01
monitor communications	0.05	0.02
staff ASOC	0	0.01
Staff TOC	0	0.01
Staff ATC	0	0.01
manage aircraft assignments	0	0.01
manage transponder	0.02	0.01
identify alternate landing sites	0.12	0.03
obtain missions	0	0.01
calculate fuel load	0	0.01

Note: In-group score contributions are 0 when only one person identified that concept. When compiling the In-group score, the number of instances was reduced by one before normalization so that participant is not contributing to the average. For cross group scores, all concept instances from the opposite specialty are counted, as no participant contributes to the cross group calculated average.

The same process was repeated on the second worksheet for vertical concepts. Function concepts were concatenated with phase, and the process was repeated on those combined statements to support horizontal comparison. Once all tables were available, the MSAccess database was used to make three queries taking the original semantic list and horizontal and vertical list, outer joining these tables with the respective scoring contribution tables (e.g. HF Closed Vertical, SY Closed Vertical) for each Specialty, Problem type and DV. The query was designed using the MSExcel summation (Σ) function to sum the In-group and Cross-group scores, and to group the sums by Participant, Model domain type and Problem Type. The vertical score join query from the MSAccess query builder provides an example, as depicted in Figure 10.

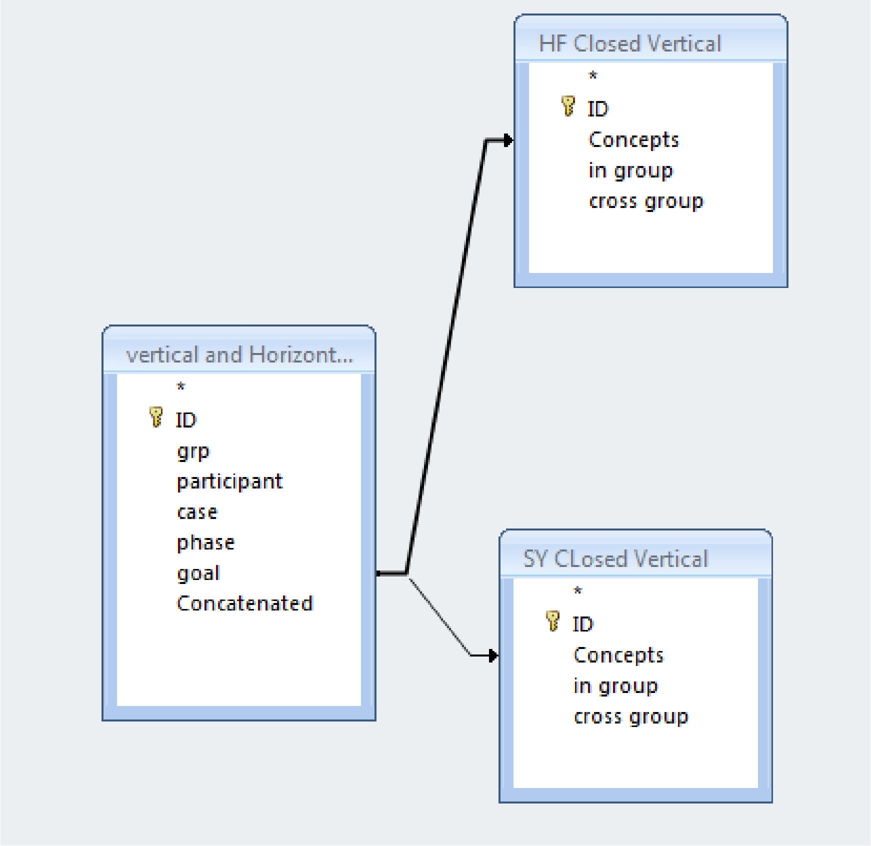

Figure 11. Scoring query in the MSAccess database. This query lists all of the “goals” in the table of vertical and horizontal concepts matched to concepts and their score contribution for In-group and cross group domains. These tables are combined via an outer join.

The query results were exported back to MSExcel. This export worksheet listed In-group and Cross-group scores by specialty type, summed for each participant at both levels of complexity. The query was not designed to differentiate between specialty type and a model domain (I.e. to determine that HF were In-group when compared against other HF and cross group when compared against SY), and thus resulted in some "nonsensical" scores. The nonsensical scores (i.e. HF "cross group" for an HF type participant) were thrown out and the rest combined into the final input to the Statistical Package for the Social Sciences (SPSS19) tool. Table 7 depicts an example of the database output.

Table 7
Example of Output from the MSAccess database

grp	participant	HF Closed Vertical.In-group	HF Closed Vertical.cross group	SY CLosed Vertical.In-group	SY CLosed Vertical.cross group
SY	7	0.02	0.03	0.09	0.05
SY	7	0.04	0.06	0.06	0.03
SY	7	0.00	0.01	0.03	0.02
SY	7	0.00	0.00	0.09	0.05
SY	7	0.16	0.20	0.09	0.05
SY	7	0.00	0.01	0.09	0.05
SY	7	0.00	0.02	0.14	0.07
SY	7	0.00	0.00	0.03	0.02

Note: Scores listed in grey are nonsensical and were not used to create the final SPSS data inputs. GRP represents educational specialty, composed of Human Factors (HF) and Systems Engineering (SY) students.

Data Analysis

A 2x2x2 between and within repeated measures multivariate analysis of variance (MANOVA) was run on three dependent variables: Horizontal Agreement, Vertical Agreement and Semantic Agreement. The Independent variables were education/practice specialty (between), model domain type (within) and problem type (within). The analysis did not evaluate the impact of education/practice specialty directly, but evaluated potential interaction effects among specialty and the other IVs to control for effects of the education specialty not germane to the study, i.e., the *efficacy* or performance of participants within their education specialty. The interest of the study was on differences just resulting from *being* in a different specialty.

The usable data between the two educational specialties yielded unequal samples sizes for specialty (N=7 for HF and N=12 for SY). Although two HF and four SY models scored extremely low (<0.10) for all DV, the researcher did not discard any complete data based on anomalous values. The researcher chose to leave these scores because they were not isolated to a single participant. The principal tests compared problem type and domain model, which were completely within, with N=19. The removal of data did slightly affect counterbalancing; the net was 11 participants that started with the closed problem first and eight participants with the open problem.

Levene's test was not significant for all variable combinations, thus equality of variance is upheld. Boxes' test for equality of covariance could not be computed, because the resulting M matrices were unstable (containing negative values); therefore equality of covariance is not assumed. Because of small sample sizes and inequality of covariance, the more conservative Pillai's trace was used over Wilk's criterion for multivariate tests. Pillai's demonstrated significance for model domain, F (3, 15) = 75.08, $p<.001$, partial η^2 =.94, meaning that the DV are measuring discrete effects. For or problem type, F (3, 15) = 7.65, $p = .002$, partial η^2=.61, and the more modest values demonstrating a higher degree of covariance[1]. Pillai's for the 2-way interaction of model domain and problem type was also significant, F (3, 15) = 10.89 $p < .000$, $\eta^2 = .69$. Maunchly's test of sphericity did not present results because all IV were measured at two levels, so sphericity is assumed. Complexity certainly may exist along a continuum, but it was not measured thusly in this experiment.

Concomitant interaction effects of specialty with all three of those potential effects were not significant, implying that group differences between HF and SY classes other than those measured in the experiment did not have an effect, which was the desired finding to uphold any other results. Pillai's for these tests were: main effect of educational specialty, F (3, 15) = 0.52, $p = .672$, partial η^2=.0.10, model domain and specialty F (3, 15) = .83, $p = .497$, partial η^2=. 0.14, problem type and specialty, F (3, 15) = 7.65, $p = .301$, partial η^2=. 0.21, and the 3-way interaction of model domain, problem type and specialty, F (3, 15) = 2.11, p =.142, partial η^2=0.30.

Follow-on univariate tests did yield five significant results: all three DV for main effects of model domain, and vertical agreement for main effect of problem type and the 2-way interaction of problem type and model domain. Results of univariate tests are summarized in Table 8. These results are discussed in-detail below. Results of education specialty interaction effects are listed in Table 9.

---

[1] Some level of covariance is expected as the function concepts measured in semantic agreement are developed in conjunction with the goal concepts measured for horizontal and vertical agreement, and thus they cannot be truly Markov independent.

Table 8.
Summary of MANOVA results.

Hypothesis	df	df_{error}	Univariate *F*	*p*	η^2
H1: Main Effect of Model domain					
DV1: Horizontal Agreement	1	18	74.02	<.001	0.81
DV2: Vertical Agreement	1	18	211.34	<.001	0.93
DV3: Semantic Agreement	1	18	51.35	<.001	0.75
H2: Main Effect of Problem Type					
DV1: Horizontal Agreement	1	18	1.18	0.29	*0.07*
DV2: Vertical Agreement	1	18	18.90	<.001	0.53
DV3: Semantic Agreement	1	18	2.48	0.13	*0.13*
H3: Interaction Effect of Model Domain and Problem Type					
DV1: Horizontal Agreement	1	18	0.49	0.49	*0.03*
DV2: Vertical Agreement	1	18	18.79	<.001	0.53
DV3: Semantic Agreement	1	18	1.11	0.31	*0.06*

Note: Significant findings are highlighted in grey

Table 9.
Checks on interaction effects from specialty

Additional Statistical Checks of Specialty Interaction Effects	df	df_{error}	Univariate F	p
H1: Interaction Effect between Specialty and Model domain				
DV1: Horizontal Agreement	1	18	0.39	0.54
DV2: Vertical Agreement	1	18	0.25	0.63
DV3: Semantic Agreement	1	18	2.58	0.13
H2: Interaction Effect between Specialty and Problem Type				
DV1: Horizontal Agreement	1	18	0.00	0.10
DV2: Vertical Agreement	1	18	1.17	0.30
DV3: Semantic Agreement	1	18	1.32	0.27
H3: 3-way Interaction Effect between Specialty, Model domain, and Problem Type				
DV1: Horizontal Agreement	1	18	0.03	0.86
DV2: Vertical Agreement	1	18	2.91	0.11
DV3: Semantic Agreement	1	18	1.97	0.18

Note: Significant results would have indicated a need for additional caution required in interpreting results from Table 8 above.

Hypothesis 1. The first research question, “Is the level agreement in participant models affected by whether they are compared against a weighted average model calculated from their own specialty (In-group) or from the opposing specialty (Cross-group)?”, is addressed by an evaluation of main effects of the model domain IV on the three DVs (horizontal, vertical and semantic agreement). Means and standard deviations for all three DV for the model domain comparison are provided in Table 10.

Table 10

Means and standard deviation for HF and SY Specialty in the Model Domain Comparison

DV	Model Domain	*Mean*	*Std Dev*
Horizontal	in	0.35	0.24
Horizontal	cross	0.05	0.06
Vertical	in	0.54	0.22
Vertical	cross	0.10	0.11
Semantic	in	0.47	0.27
Semantic	cross	0.20	0.13

Note: Significant main effects are highlighted in grey.

The univariate follow-up test exhibited significant findings for H_{1a} - horizontal agreement between in and Cross-group scores by specialty, F (1, 18) = 74.02, $p<.001$, partial η^2= 0.81, for H_{1b} - vertical agreement, F (1, 18) = 211.34, $p<.001$, partial η^2= 0.93, and for H_{1c} semantic agreement, F (1, 18) = 51.35, $p<.001$, partial η^2=0.75. For each of the three measures of Agreement, the higher agreement between participants occurred when calculated with respect to an average model from the same domain (in-group) rather than the other domain (cross-group). The very high partial eta squared values demonstrating that the in-group vs. cross-group treatment was responsible for most of the observed variance. The last result (semantic agreement) was a surprise as the researcher expected to fail to reject H_{1c0}. This was likely due to an anomaly with the SY specialty and not differences incurred at the open level as was originally expected.

As described earlier, the data also were examined for potential interaction effects with specialty. Interaction effects with specialty and model domain were not significant so there are no additional confounding effects from specialty. The observed means for cross group reading decreased from the In-group scores for all three DV. All of the scores for both specialties are tightly banded, with apparently similar variance, for all DV, except that the SY In-Group scores for semantic agreement. This presents an interesting anomaly, which might be caused by the greater variety in education and industry experience for the SY class over the HF class providing participants. Figures 12, 13 and 14 illustrate this.

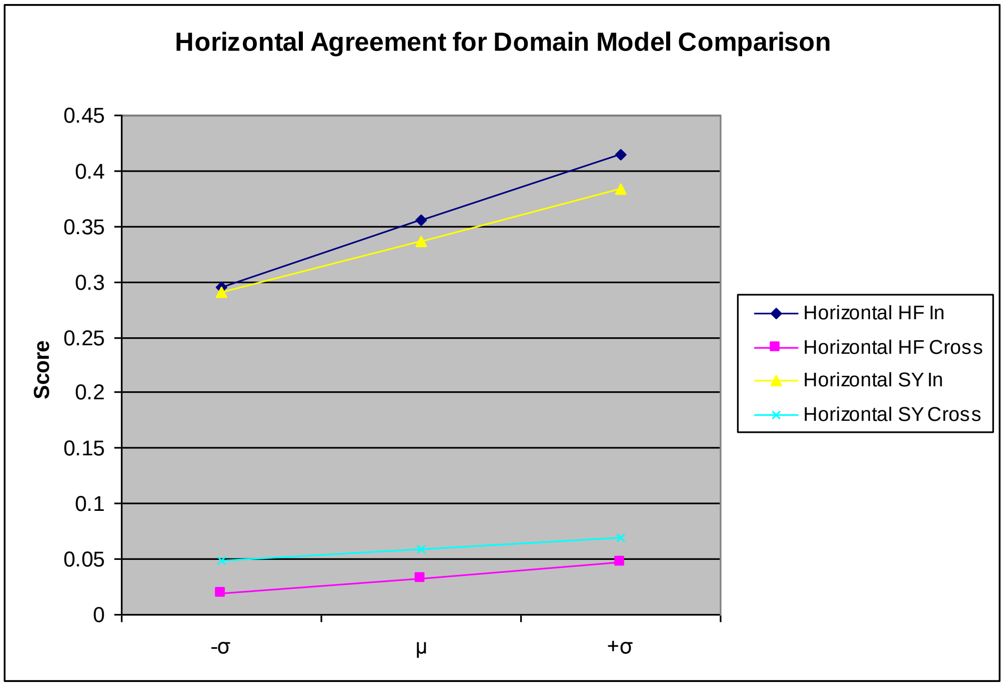

Figure 12. Means and variance for domain model comparison for the horizontal agreement DV. In and Cross domain scores are neatly clustered and visually different, with narrow variance bands.

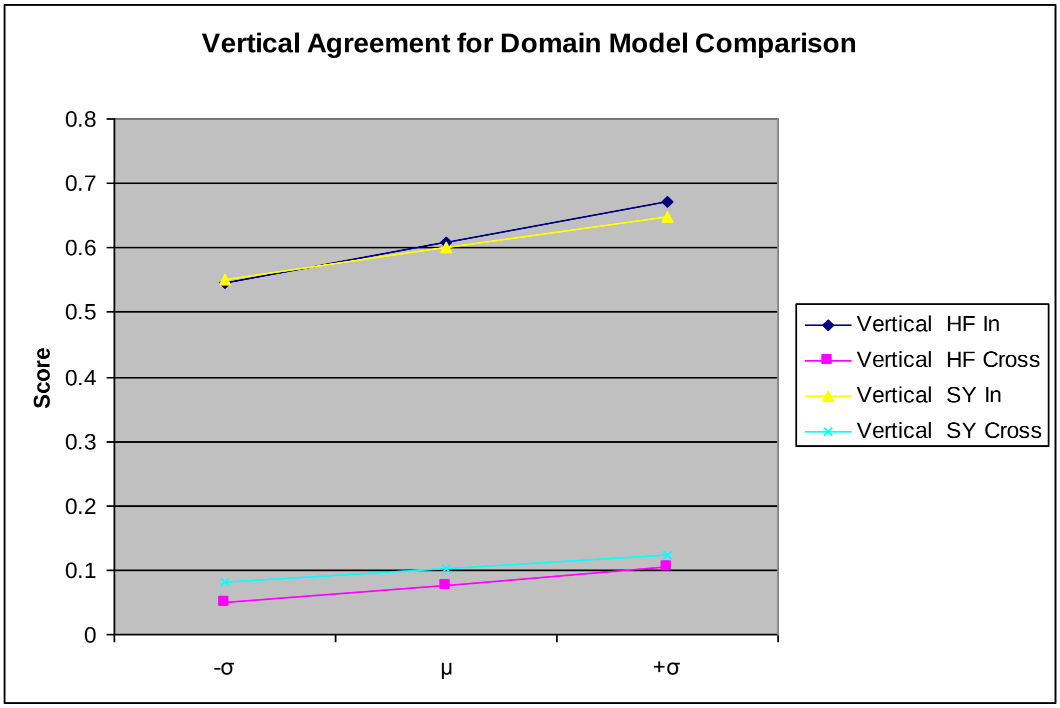

Figure 13. Means and variance for domain model comparison for vertical agreement DV. Vertical agreement displayed the tightest band of similarity between the two specialties.

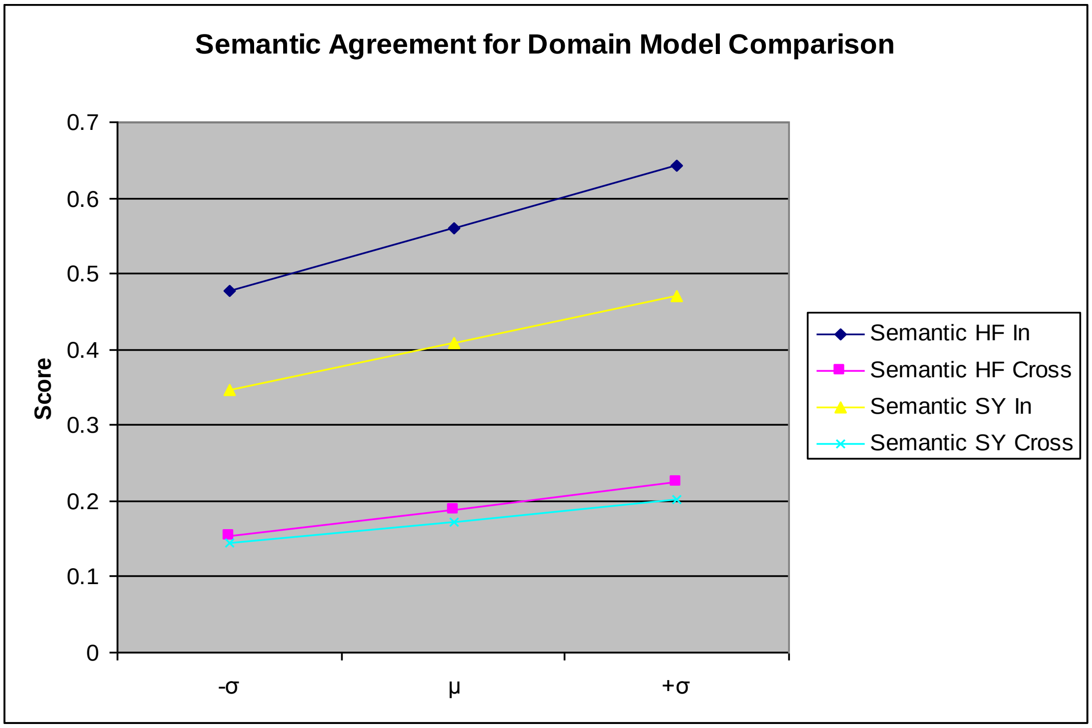

Figure 14. Means and variance for domain model comparison for semantic agreement DV. The drop in SY scores vs HF scores presents an interesting anomaly.

The differences in the two "average" models which yielded the in-group vs. cross-group differences become readily apparent when viewing frequency histograms of common concepts. Figures 15 and 16 display the frequency of most common concepts identified by SY participants, compared against the incidence for HF participants in the same problem case. In both cases, the obvious void presents the source of the model domain comparison difference. A histogram comparing frequency of concepts identified by HF and SY students in the closed problem case is depicted in Figure 15, and a separate plot for the open problem case is depicted in Figure 16 (overleaf).

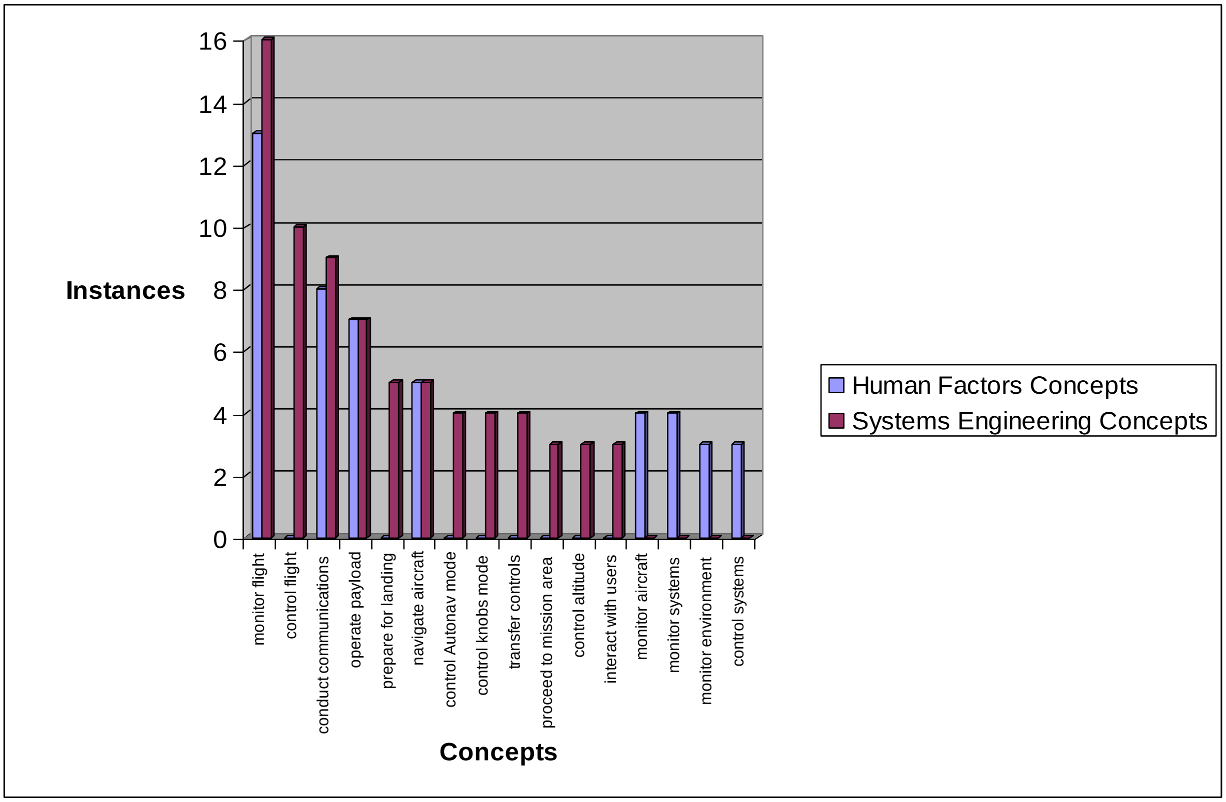

Figure 15. Histogram of vertical concepts identified for the closed problem case, compared for human factors and systems engineers (where more than two SY students recognized the concept)

As seen in Figure 15, Systems engineers overwhelmingly focused on monitoring and controlling the entire flight as the chief automated functions of the GCS. They further identified a number of specific technology controls, which were infrequently identified by HF participants, or not at all. SY participants chiefly identified the technology features of the solution. Conversely, HF students identified monitor aircraft (a user centered function, as well as other functions associated with user situation awareness (monitor environment, monitor aircraft, monitor systems).

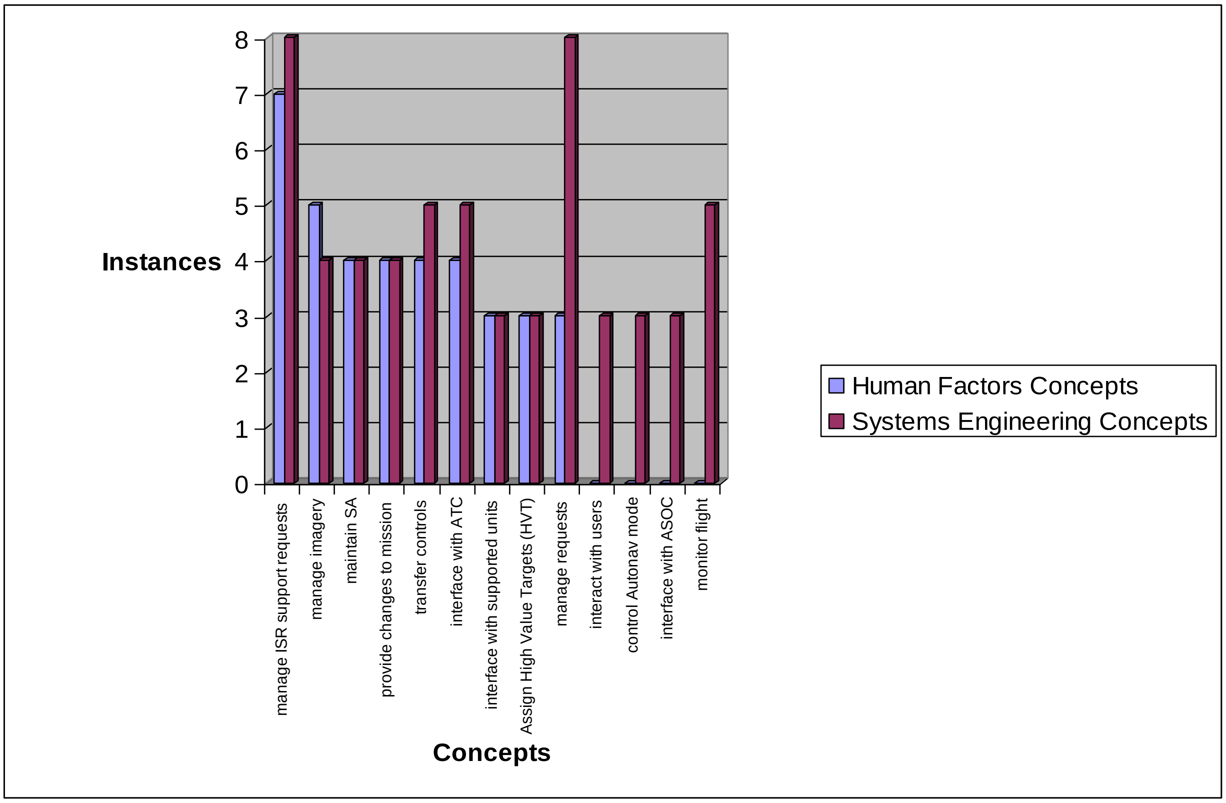

Figure 16. Histogram of vertical concepts identified for the open problem case, compared for human factors and systems engineers (where more than two SY students recognized the concept)

Hypothesis 2. The second research question, "Is the level agreement in participant models affected by the complexity of the system under consideration?" has two associated hypotheses. The first, hypothesis $H_{2,}$ looked at semantic, horizontal and vertical agreement as potentially impacted by the main effect of Problem type. Table 11 presents the means and standard deviations for participants for all three DV when compared against open and closed problem types.

Table 11

Means and standard deviation for all DV compared against problem type

DV	Problem Type	*Mean*	*Std Dev*
Horizontal	Open	0.18	0.20
Horizontal	Closed	0.23	0.26
Vertical	Open	0.27	0.23
Vertical	Closed	0.37	0.32
Semantic	Open	0.29	0.26
Semantic	Closed	0.38	0.24

Note: Significant main effects are highlighted in grey.

The follow-up univariate test indicated significant results only for vertical agreement F (1, 18) = 13.9, $p<.001$, η^2= 0.53. Neither horizontal nor semantic agreement evidenced significant difference (See Table 8). Again, the interaction effect of specialty with problem type was likewise not significant, suggesting that the vertical finding depicts a legitimate difference. In other words, when agreement was calculated in the vertical manner, higher mental model agreement occurred on simple/closed problems than on the more complex/open problems regardless of both education/practice specialty and in- or cross- group analysis,

Examination of the variance bands associated with the horizontal and semantic agreement scores demonstrates the reason for non-significance. For horizontal agreement, the In-group scores were noticeably noisier that the cross-group scores and this noise overcame any mathematical difference resulting from problem type. In the horizontal case, the variance is demonstrably greater for the in group case than the cross group case, this is because the concatenated strings of goal and phase used to compare horizontal agreement would amplify any slight changes in the identified goal concepts, measured alone in the vertical case.

Semantic agreement in the problem type comparison exhibited an additional anomaly, where all the scores and variance bands were narrowly clustered around the same value, except for SY open scores, which were much lower than the others were. This finding suggests that the

HF students might be more consistent when dealing with ambiguity. This experiment did not attempt to measure whether they could handle it well, but this finding suggests that ambiguity does not negatively affect their performance more than the less ambiguous problem. Figures 17, 18, and 19 depict these results.

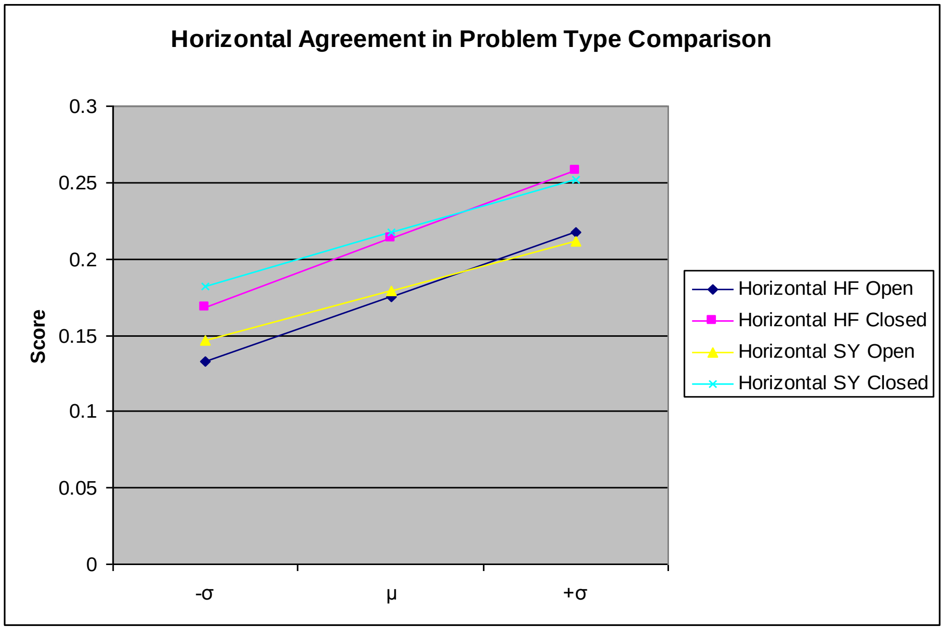

Figure 17. Means and variance for horizontal agreement in the problem type comparison. The relatively larger bands of variation (noise), coupled with the small shift in means did not exhibit statistically significant difference.

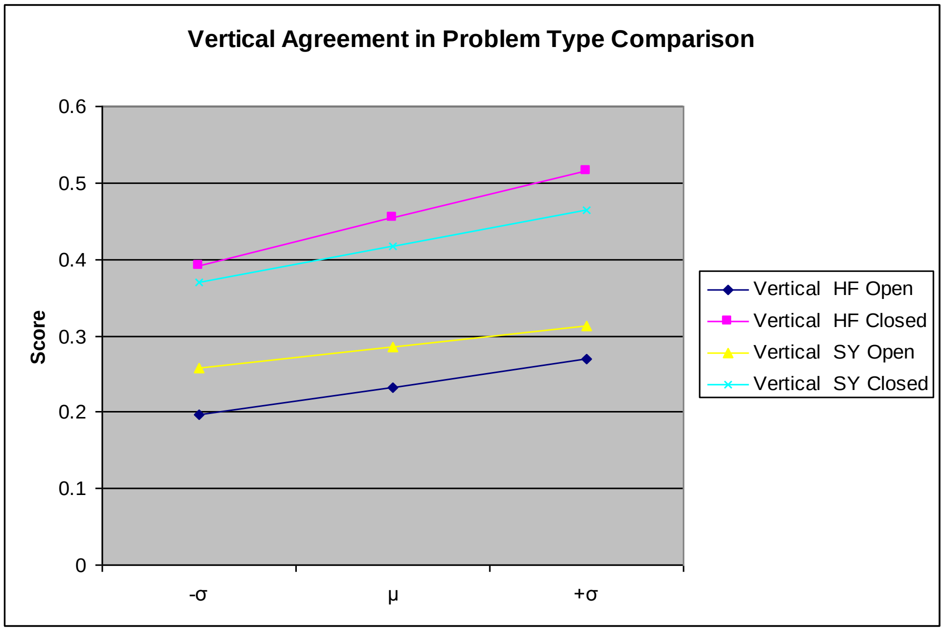

Figure 18. Means and standard deviation for vertical agreement in problem type comparison. These exhibited significant difference.

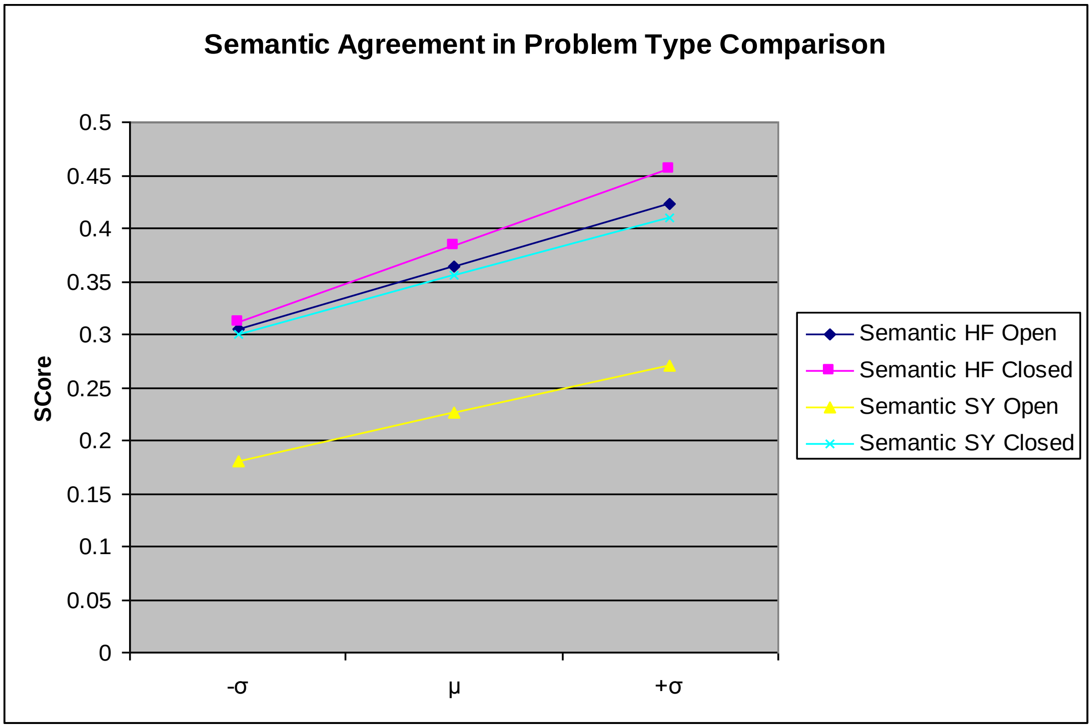

Figure 19. Means and standard deviation for semantic agreement in problem type comparison. The narrow cluster of scores around 0.4, along with anomalous values for SY prevented a significant difference.

Hypothesis 3. The last hypothesis, also tied to the second research question regarding the impact of complexity, sought to find evidence that the level of complexity would introduce even more differences in agreement between individual models than that due to being from different educational domains. This was examined by a 2-way interaction between model domain and problem type. Table 12 lists means for the possible interactions.

Table 12
Means and standard deviation for the two-way interaction effect of model domain and problem type

DV	Problem Type	*Mean*	*Std Dev*
Horizontal	In Open	0.31	0.19
Horizontal	Closed Cross	0.06	0.06
Horizontal	Cross Open	0.05	0.06
Horizontal	In Closed	0.39	0.28
Vertical	In Open	0.47	0.16
Vertical	Closed Cross	0.13	0.13
Vertical	Cross Open	0.08	0.07
Vertical	In Closed	0.62	0.24
Semantic	In Open	0.43	0.10
Semantic	Closed Cross	0.26	0.14
Semantic	Cross Open	0.14	0.10
Semantic	In Closed	0.50	0.25

Note: Significant differences are highlighted in grey

This test exhibited significant results in the vertical agreement score *F* (1, 18) = 18.79, $p<.001$, $\eta^2 = 0.53$. Horizontal and Semantic agreement scores were not significant for this interaction. The three way interaction effects of specialty, model domain and problem type for all

DV were likewise not significant, suggesting a valid difference for vertical agreement. Figures 20, 21, and 22 depict these results.

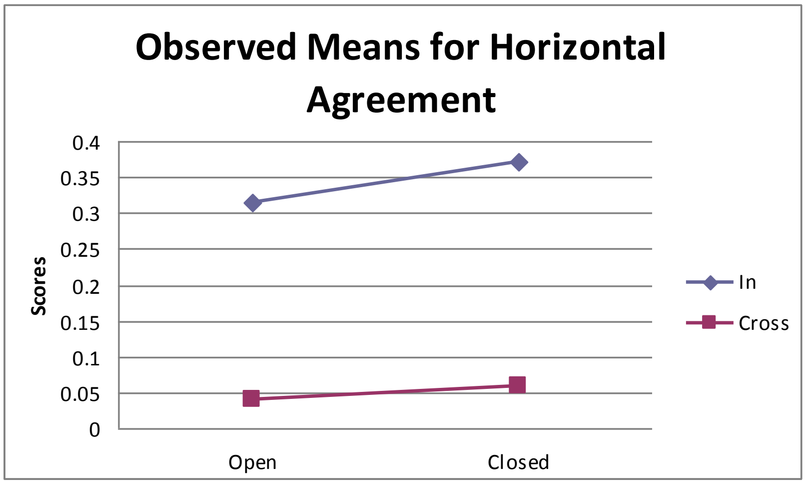

Figure 20. Observed Means for Horizontal Agreement. Cross Domain scores are lower in all cases. The nearly parallel lines indicate a NOT statistically significant interaction effect.

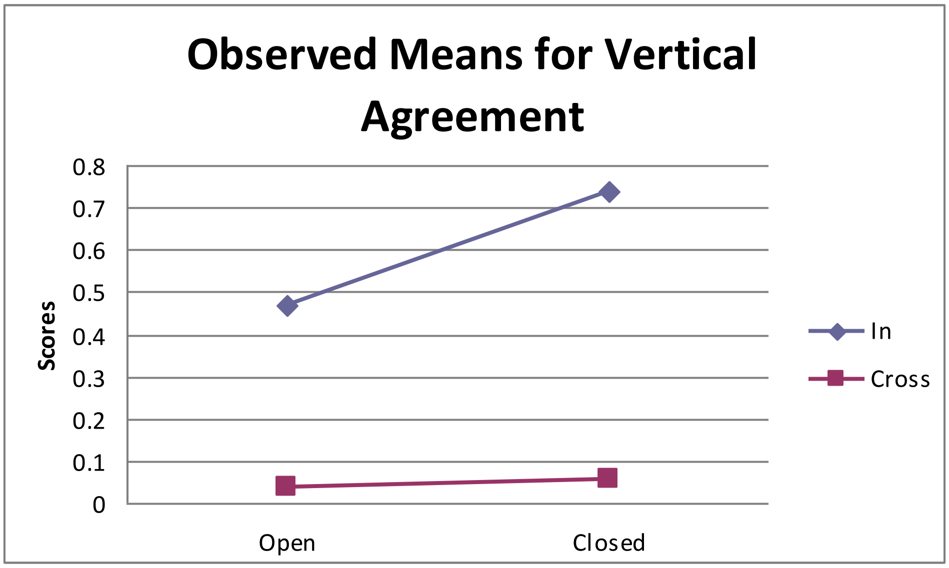

Figure 21. Observed Means for Vertical Agreement. Cross Domain Scores are lower than in domain scores. The closed problem had higher agreement, and there IS a statistically significant interaction effect between Problem type and Model domain.

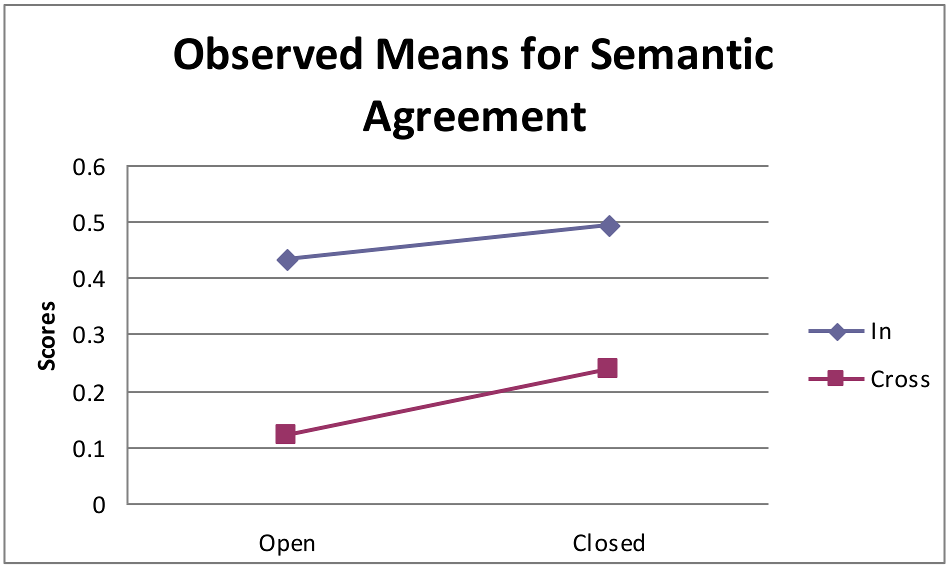

Figure 22. Observed means for semantic agreement. Scores for the cross domain are lower than in domain, but the parallel lines did NOT exhibit a significant interaction effect.

Post Hoc Analysis. Looking at figure 21, it appears that when agreement was calculated cross-group, agreement was always lower than when calculated in-group (regardless of problem type). Additionally, it appears that the highest agreement came when comparing individual's mental models to an in-group average regarding a simple/closed problem. These apparent differences were examined using *post-hoc* paired t-tests on all interaction means combinations for the vertical agreement DV. The complete results of the t-tests and Fischer's Least Significant Difference (LSD) are listed in Table 13 (overleaf). SPSS would not generate the LSD automatically, so they were calculated manually in MSExcel.

Post Hoc tests provided clear support for the two-way interaction of problem type and model domain. As seen in Table 13, all pair comparisons met the LSD criteria except for Pair 2, Cross-Closed and Cross-Open (indicated in Figure 21 with the nearly flat purple line) and Pair 3, In-Open Cross-Open, which was the closer pair of means in the same Figure.

Table 13

M, SD and LSD for post hoc t-tests for the interaction between Domain and Problem type

	Paired Samples													
Comparison	Domain	Problem Type	Group Mean	Domain	Problem Type	*Group Mean*	*Std Dev*	*Cor.*	*N*	*t*	*p*	*MSE*	*LSD*	*Actual Dif*
Pair 1	In	Closed	0.74	In	Open	0.47	0.26	0.23	19	-5	<.001	0.02	-0.22	0.27
Pair 2	Cross	Closed	0.12	Cross	Open	0.06	0.14	0.03	19	1.9	0.08	0.03	0.10	0.06
Pair 3	Cross	Open	0.06	In	Open	0.47	0.14	0.58	19	13	<.001	0.02	0.61	0.41
Pair 4	Cross	Closed	0.12	In	Closed	0.74	0.23	0.4	19	12	<.001	0.02	0.57	0.62
Pair 5	In	Closed	0.74	Cross	Open	0.06	0.23	0.36	19	13	<.001	0.01	0.47	0.68
Pair 6	In	Open	0.47	Cross	Closed	0.12	0.22	-0.1	19	5.9	<.001	0.01	0.22	0.35

Note: Tests meeting the LSD criteria are highlighted in grey.

Questionnaire data. Twenty Eight students provided responses to the post exercise questionnaire, including nine (3 HF and 6 SY) of the participant concept data ultimately included in the main experiment All the responses defined "system" more or less with the canonical "integrated set of elements that accomplish a defined objective" (International Council on Systems Engineering, 2004). The responses to other definitions, however, exhibited a great deal of variety. Three students responded with the canonical INCOSE definition of function as a process of matter, energy or information, and none of those individuals provided a complete usable data set. Of the three students that described "complexity" as relating to information content, two provided usable data, one HF and one SY. Students who tied the definition of function to solutions were all non-native English speakers (Chinese), so it is possible this is tied to a translation error. The main experiment did not use their concept data. Most participants identified one or both "mission" and "function" as goal oriented, but half of the participants provided ambiguous definitions for task and activity with respect to their definition of function. It was interesting that almost all of the students defined "capability" in terms of chance at achieving requirements, when the Department of Defense defines it as a user-facing discussion of value, and Suh (2001) defined “complexity” in term of “chance to achieve requirements.” Lastly, approximately one third of participants defined complexity in terms of difficulty. The responses summary is in Table 14.

Table 14

Responses to the Post Exercise Questionnaire

Requested Definition	General Category of Answer	Number of Responses
Capability	ability to meet requirements	22
	other	5
Complexity	difficulty	7
	number of interactions	11
	information content	3
	other	7
Mission/Function	goal oriented	22
	tied to solution	3
	canonical definition	3
Task/activity	tied to work over time	7
	ambiguous with mission/function	14
	other	7

Note: Out of 28 submissions, all were able to define “system” more or less canonically, but the other definitions evidenced a lot of variation.

Problem scope comparison. Most of the students did not answer the demographics question regarding the comparative scope of the open and closed problem statements, although the average of the seven who did was 3.2 on a scale of 1(dissimilar) to 5 (identical). Problem difficulty was not to be considered, just scope, as the open problem was intended to be “harder,” and based on anecdotal feedback, the students who did respond may have confused scope for difficulty. Instead, a comparison of total concepts for the closed and open problems controlled for problem scope as a potential confound. Calculating the number of concepts per mission phase (the problem statement provided phases) yielded averages of 5.5 concepts/phase for the closed problem vs. 4.7 concepts/phase for the open problem and performing a one way analysis of variance (ANOVA) test on the concepts per branch for all the participant data yielded no significant results *F* (1, 18) =0.63, *p=.432*, providing evidence for a null hypothesis that the scope of the two groups are not different.

Boundary and level ambiguity. Following the reduction of data in the main experiment, the researcher compared the concept list for crossovers between the function and sub-function level, and for cross over between the open and closed problem cases. This analysis gave insight into the degree of ambiguity in understanding the boundaries between the level of abstraction, and between the responsibilities of the two hypothetical systems (deliberately designed to be complementary). Comparison of concept overlap between the function and sub-function worksheet demonstrated that 31.7% of the concepts defined at the top level by one participant were identified at the sub-function level by a different participant, with the most common occurrences depicted in Figure 23.

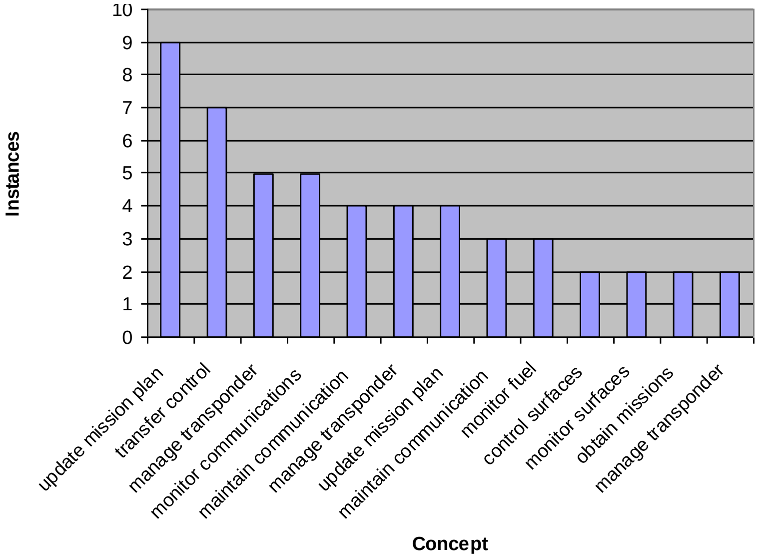

Figure 23. Histogram of most common occurrences of model concepts that were identified at ambiguous “function or goal" and "sub function" levels by different participants.

Comparison of concepts for both function and sub-function identified during the closed problem with those identified during the open problem yielded similar results. This comparison found that 19.5% of the concepts defined by one participant in the closed problem were defined by another participant in the open problem with the most common occurrences depicted in Figure 24.

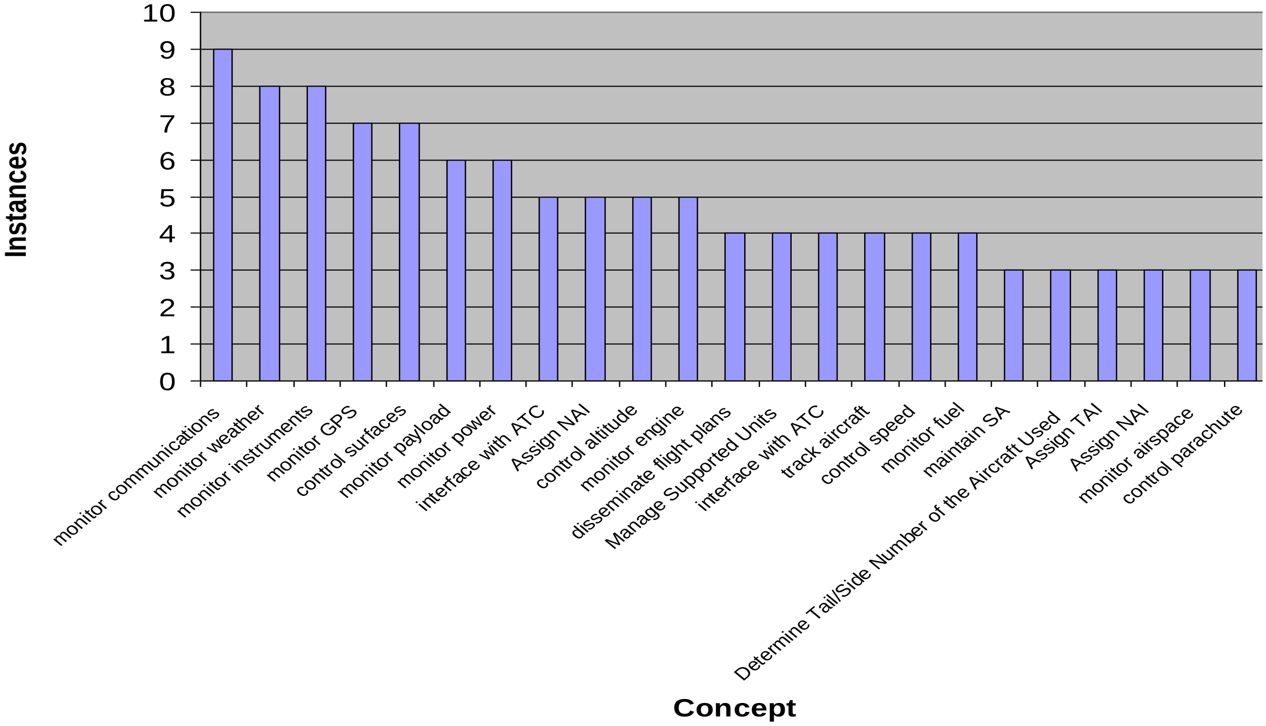

Figure 24. Histogram of most common occurrences of model concepts that were identified at ambiguous system boundaries between the two hypothetical systems defined in models created for the open and closed problem statements by different participants.

Discussion

The purpose of this study was to examine the effect of complexity and training (as a human factors or systems engineer) on the problem domain (as opposed to task) component of the mental model of a design engineer. These questions were evaluated with three IVs: specialty (degree program of the participants), comparison model domain (i.e. against the same specialty or from the opposite specialty) and problem type (representing complexity level), each looking at possible effects on three measures of mental model agreement. The three measures assessed the similarity between participant models and a weighted group average model for leaf node "sub-functions" (i.e. semantic agreement), parent node "goal/functions" (i.e. vertical agreement) and phase/goal couples (i.e. horizontal agreement). The experimenter evaluated three sets of hypotheses, each with three subordinate hypotheses (one for each DV). These hypotheses looked at the effects of model domain (comparing against the average model from the same educational specialty, or the opposing educational specialty), problem complexity, and the combined effect of both, on mental model agreement. The analysis evaluated the potential main effects of domain model and problem type, as well as the two-way interaction between them. Additionally, I conducted a cross check of the two-way interactions of specialty with model domain and problem type, as well as the three-way interaction of specialty, problem type and model domain. This check controlled for performance differences between HF and SY not considered part of the experiment. This experiment was examining the effect of *being* in a group, not whether one specialty was more or less effective or participants of that specialty more or less synchronized in their thinking than the other.

The current study's evidence for differences in human factors and systems engineering perspective in describing the same problem is pretty convincing, as all three DV exhibited significant difference in respect to model domain (comparison of In-group and Cross-group scores). I was surprised that semantic agreement exhibited such a large difference (observed means of 0.49 vs. 0.23) in the closed case, as all of the required functions were provided in the problem description, but it is possible that the specialty differences focused on different elements of the problem statement as being relevant.

Similarly, in examining the differences in mental models in respect to closed vs. open problems, demonstrating significance for vertical agreement provided some evidence to suggest that the complexity of the problem does increase the variation between team members. The effect of the two-way interaction for vertical agreement further suggests that the specialty of the team members will exacerbate the problem of mental model synchronization. In the case of horizontal agreement, all of the overall scores were much lower, since the horizontal was a concatenation of two concepts; there were twice as many chances for differences to appear. The Semantic scores were clustered together, with the exception of the SY open scores. It is possible this was due to the problem statement, the HF students generally left in the ambiguous statements (being more comfortable with ambiguity), whereas the systems students tried to actually specify detailed functions, and they were more likely to diverge (which was the intent of the open problem).

I would conclude that the level of noise increase (increase in standard deviation of scores), while making statistical significant findings difficult, is of itself an important finding with respect to synchronizing team mental models. It is especially interesting that the variance of the In-group scores was so much larger than that of the cross group scores (graphically, albeit not in a statistically significant way) suggesting that even within a homogeneous team that communication is hard.

In my work as an engineer, I have observed that ambiguity in the goals/means decomposition causes many of the communication difficulties in team settings. In my work, this appears as inconsistent use of words such as “subsystem” or “component.” These ambiguities arise from differences in individual perspective as Lintern cited (Lintern, 2009); one person's "goal" is another person's "means", depending on their status within the organization.

Additional difficulties arise from ambiguity in describing the boundary defining "what is the system?" vs. "what is the environment?" The problem statement attempted to control for level ambiguity by providing the first two levels of decomposition, and by defining interfaces and information exchanges in the body of the problem description. The ambiguity of boundary was intentional, however, in the complex case, as open, ill-defined and dynamically changing boundaries is one of the salient characteristics of a complex system

As suggested by Defranco et al. (2011), implementing a framework to synchronize the task component of the team mental model has an impact on improving the similarities between the problem domain components of individuals working in a team. The guest lecture was designed to control for task component of the mental model, but this control could and should be improved upon.

Additionally, this experiment assumed that the problem domain component of the team mental model was a summation of the constituent parts of individual problem domain mental models, disregarding the impact of team work or emergent discovery as part of the team process. The agreement scores evaluated the Hamming distance of an individual model from the union of all models. A real team composed of participants of a specialty would exhibit some emergent discovery and pairing down processes not present in simple summation of individuals' work. Team dynamics also include alpha personalities, social loafers, and other idea culling and fusion factors, which would generate different results than a summation of individuals approach.

The use of students, even post-graduate level students, as participants imposed some limitations on the study. The manifest floor effect, where only 1/3 of the students could properly perform the task, had the concomitant effect of reducing the participant pool. Use of more experienced practitioners, and a larger sample size, might likely have changed the results, and hopefully made some differences even clearer. Initially, XMind was chosen to prevent potential confounds from tool type, although ultimately, it turned out that all the participants had experience with using CORE. The use of the more structured FFBD features of CORE might have provided additional control for the task component of the individual mental models through its use of validation logic and strongly coded (in the human factors sense of the word) user interface. All of these factors likely contributed to the noise evidenced in the study.

If I pursued a second study, I would control for the variation in the specification of concepts, and instead focus on the effects of boundary on the participants understanding of the design problem. This future study would define as part of the problem statement a candidate set of low-level functions for the students to select and organize into the two provided problem cases and the 7 mission phases. This study would measure Semantic differences explicitly in terms of inclusion or exclusion within the system boundary, (i.e. the participants would not have to use all the provided concepts, only the ones they though appropriate). This design would then be especially valuable when comparing problem type levels. Moreover, it would provide greater control for goals/means ambiguity by “fixing” the level of detail in the sub-functions. Most importantly, it would avoid the floor effects that affected this current study, in which only about 1/3 of the participants could perform the task adequately. Additional studies could investigate the impact of goals/means ambiguity and their relationship to boundary.

An additional enhancement to the study would include the use of multiple teams, instead of individuals, as the participants. While this would require an even larger number of participants, having teams of three would allow for some team dynamics at play, and probably level out some of the individual differences, as the team members would "regress towards the mean," as part of their team brainstorming session.

I believe this study presented a unique contribution. The Defranco study addressed the measurement of mental models in engineering design teams (albeit without calling them such), but this study combined this direct measure with the effects of complexity, and looked directly at the challenges of integrating multi-disciplinary teams. Similarly, this study focused on a relatively new problem domain, that of unmanned aircraft systems (UAS), which are stressing the state of the art processes and tools for both systems engineers and human factors engineers. UAS present interesting problems, which will require innovative solutions, which in turn require a solid understanding of the root causes.

In conclusion, this research found some convincing evidence to suggest that the differences in training and perspective between human factors engineers and systems engineers is such that they conceptualize the same design problem in different ways. This finding provides additional validation of the Joint Cognitive Systems model presented in Figure 2, although much more work needs to be done with respect to understanding the nature of the human/machine intersection space in the model. The Human Factors engineers tend to think of a prospective system as a user facing tool, while the systems engineers tend to view it in terms of the technology. I further believe that continued research into the effect of complexity, especially by continuing this line of inquiry with more experienced practitioners, could reap rewards to future engineering projects of complex socio-technical systems.

The impact on industry seems obvious, communication is essential for success. Proper communication requires both the sender and receiver to have the same understanding of the message. Getting multi-disciplinary teams to agree on the purpose and solution is a critical part of success for a development program, and any program with geographic separation will require teams to work in partial isolation to return later for integration. Successful integration absolutely requires that these teams have a common understanding of the problem at hand. Failure to achieve this common understanding is likely to drive cost and schedule, and potentially signal failure of the entire program.

I believe a part of the solution is in the development of common modeling languages to be shared between systems and human factors engineers. I pursued an independent study as part of this degree program to that end, calling it the Cognitive Systems Engineering Modeling Profile (CSEMP). It extended the Systems and Unified Modeling Language (SYSML/UML) to include concepts used by CSE, providing a translational mapping between one specialty and the other. Most importantly, its intent was not to come up with a universal language, but rather to make explicit where discontinuities between the two perspectives existed, so that they could be addressed and resolved quickly during the development process.

References

Albrechtsen, H., Andersen, H., Bødker, S., & Pejtersen, A. (2001). Affordances in activity theory and cognitive systems engineering. Riso National Laboratory: Roskilde Denmark. Retrieved 14 Apr 2011, from http://130.226.56.153/rispubl/SYS/syspdf/ris-r-1287.pdf

Avnet, M. (2009). *Socio-cognitive analysis of engineering systems design: shared knowledge, process, and product*. Unpublished doctoral dissertation, Massachusetts Institute of Technology. Retrieved 26 November, 2010 from http://esd.mit.edu/people/dissertations/avnet.pdf

Bartolomei, J., Hastings, D. , Neufille, R., & Rhodes, D. (2012) Engineering systems multiple-domain matrix: An organizing framework for modeling large-scale complex systems. *Systems Engineering, The Journal of the International Council on Systems Engineering 15(1)*, 41-59.

Bisantz. A. M., & Catherine, B.M., (eds). (2009). *Applications of cognitive work analysis*: CRC Press. Retrieved 19 April 2011, from http://www.crcnetbase.com.ezproxy.libproxy.db.erau.edu/doi/pdfplus/10.1201/9781420063059.ch13

Buede, D. (2000). *The engineering resign of systems: Models and methods*. Canada: Wiley and Sons.

Cannon-Bowers, J., & Salas, E. (1998). *Making decisions under stress: Implications for individual and team training.* Washington, DC: APA.

Cummings, M., Kirschbaum, A., Sulmistras, A., & Platts, J. (2006). *STANAG 4586 Human supervisory control implications.* Air and Weapon Systems Dept, Dstl

Farnborough & the Office of Naval Research, Retrieved 26 November, 2010 from http://web.mit.edu/aeroastro/labs/halab/papers/STANAG_UVS.pdf

Defranco, J., Neill, C., & Clariana, R. (2011). A cognitive collaborative model to improve performance in engineering teams: A study of team outcomes and mental model sharing. *Systems Engineering, The Journal of the International Council on Systems Engineering 14(3)*, 267-277.

Endlsey, M., & Kaber, D. (1999). Level of automation: Effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics 42(3).* 462-492. Retrieved from http://www.satechnologies.com/Papers/pdf/Endsley_Kaber_Ergo_99.pdf

Esparza, J., & Lakos, C. (2002). Application and theory of Petri Nets. Proceedings of the International Conference ICATPN 2002. Retrieved 24 Feb, 2012 from http://books.google.com/books?id=QvATm4v3zoYC&pg=PA292&lpg=PA292&dq=INCOSE+petri+nets&source=bl&ots=5viSa8C8kT&sig=iblojOhI5QJXyL03EgVm_NUrqFs&hl=en&sa=X&ei=0vRHT62cKYj9sQLa5MDXDA&ved=0CDUQ6AEwAw#v=onepage&q=INCOSE%20petri%20nets&f=false

Friedenthal, S., Moore, A., & Steiner, R. (2008). *A practical guide to SYSML, the systems modeling language.* MA: Morgan Kaufman Press.

Gordon, J., Burns, J., & Giebenrath, J. (2005). *A common data representation for HSI: The universal performance data model:* Proceedings of the 2005 Human Systems Integration Symposium

Gordon, J., Burns, J., Sheehan, J., Ricci, K. & Pharmer, J. (2005). *A framework for enabling HSI within systems engineering: HSI and the Navy's multi-mission maritime aircraft*: Proceedings of the 2005 Human Systems Integration Symposium

Hollnagel, E. (ed) (2003). *Handbook of cognitive task design*. NJ: Lawrence Erlbaum Associates.

International Council on Systems Engineering (2004). *Systems engineering body of knowledge v2*. Retrieved 20 October 2012 from http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&ved=0CDkQFjAE&url=http%3A%2F%2Fsce.uhcl.edu%2Fhelm%2FSENG_DOCS%2Fseng_hdbk_v2a.doc&ei=VPF6UOT9CZHU8wSFk4DoAg&usg=AFQjCNF4du24K9Rv2uwpjo1WxwDwP9rDdA

Johnson, C., & Holloway, C. (2006). Questioning the role of requirements engineering in the causes of safety-critical software failures. *The 1st Institution of Engineering and Technology International Conference on System Safety*. 352-361. Retrieved 8 Feb 2012 from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4123725

Johnson-Laird, P.N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness.* Cambridge: Cambridge University Press. Retrieved 26 April 2012 from University of Central Florida Library Online Collection.

Johnson-Laird, P. (2006). *How we reason*. Oxford University Press. Retrieved 26 April 2012 from http://books.google.com/books?id=UjYsJN0krNYC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Jonassen, D., Tessmer, M., & Hannum, W. (1999). *Task analysis methods for instructional design*. NJ: Lawrence Erlbaum Assoc.

Kossiakoff, A., & Sweet, W.N. (2003). *Systems engineering principles and practice.* NJ: John Wiley and Sons.

Lim, B.C., & Klein, K. (2006) Team mental models and team performance: A field study of the effects of team mental model similarity and accuracy. *Journal of Organizational Behavior* 27, 403–418. Retrieved 26 April 2012 from http://www.management.wharton.upenn.edu/klein/documents/Lim_Klein_Team_mental_models_2006.pdf

Lintern, G. (2009). *The foundations and pragmatics of cognitive work analysis: A systematic approach to developing large scale information systems*. V 1.0 Retrieved 12 Mar 2011, from http://www.cognitivesystemsdesign.net/Downloads/Title%20Contents%20Preface.pdf

Mackiewicz, R., & Johnson-Laird., P. (2011). Reasoning from connectives and relations between entities. *Memory and Cognition DOI 10.3758/s13421-011-0150-8.* Retrieved 26 April 2012 from http://mentalmodels.princeton.edu/papers/2011connectives&relations.pdf

Madni. A. (2010). Integrating humans with software and systems: Technical challenges and a research agenda. *Systems Engineering. 13(3)*, 232-245.

Militello, L., Dominguez, C., Lintern, G., & Klein, G. (2010). The role of cognitive systems engineering in the systems engineering design process. *Systems Engineering*, *13(3)*, 261-273.

Minai, A., Braha, D., & Bar-Yam, Y. (eds.) (2006). *Complex engineered systems science meets technology*. NY: Springer.

Mitleton-Kelly, E. (2003) Ten principles of complexity and enabling infrastructures. *Complex systems and evolutionary perspectives of organizations* Retrieved 23 Feb 2012 from http://www.psych.lse.ac.uk/complexity/events/PDFiles/publication/Ten_principles_of_complexity_enabling_infrastructure.pdf

Mueller, M. W. (2008). Human systems integration: What's it all about? *INCOSE Insight 11(2).* USA:INCOSE

National Transportation Safety Board. (2007). Aircraft accident report: Predator B OMAHA 10, Libby Army Airfield, Arizona, 24 April 2006, in *Technical Report NTSB-AAR-CHI06MA121*. 2007. Washington. Retrieved from http://www.ntsb.gov/safetyrecs/private/querypage.aspx

Neumann, A., Badke-Schaub, P., & Lauche, K. (2006). Measuring the development of shared mental models in product design teams. *16th World congress of ergonomics, Maastricht*. Retrieved from University of Central Florida Library Online Collection 26 April 2012.

Norman, D., & Kuras, M. (2006). Engineering complex systems. *Complex engineered systems science meets technology*. Minai, A., Braha, D., & Bar-Yam, Y. ed. NY: Springer. pp 206-245.

Osorio, C., Dori, D., & Sussman, J.(2011). COIM: An object oriented process based method for analyzing architectures of complex, interconnected, large scale socio-technical systems. *Systems Engineering, 14(4)*, 364-381.

Parasuraman, R., & Wickens, C. (2008) Humans: Still vital after all these years of automation. *Human Factors 50(3):* 511-520.

Pew, R., & Mavor, A. (ed) (1999). *Modeling human and organizational behavior*. Washington DC: National Academy of Sciences Press.

Pew, R., & Mavor, A. (ed) (2007). *Human systems integration in the system development process: A new look.* Washington DC: National Academy of Sciences Press. Retrieved 3 March 2012 from http://www.nap.edu/openbook.php?record_id=11893&page=R2

Potter, S., Elm, W., & Gualtieri, J. (2002). Case studies: Applied cognitive work analysis in the design of innovative decision support. Retrieved 5 March 2012 from http://www.resilientcognitivesolutions.com/wordpress/wp-content/uploads/2007/11/case-studies-chapter-2002-potter-et-al.pdf

Potter, S., Elm, W., Roth, E., Gualtieri, J., & Easter, J. (2001). Bridging the gap between cognitive analysis and effective decision aiding. Retrieved 12 Mar 2011, from http://mentalmodels.mitre.org/cog_eng/reference_documents/Bridging%20the%20Gap--revised2.pdf

Rasmussen, J. (1990). Mental models and the control of action in complex environments. In D. Ackermann, D. & M.J. Tauber (Eds.). *Mental models and human-computer interaction 1* (pp. 41-46). North-Holland: Elsevier Science Publishers.

Rouse, W. B. (2007). Complex engineered, organizational and natural systems. *Systems Engineering, 10(3)*, 260-270.

Sanders, M., & McCormick, E. (1993). *Human factors in engineering and design*, 7th ed. NY: McGraw Hill.

Sheard, S.A., & Mostashari, A. (2009). Principles of complex systems for systems engineering. *Systems Engineering. 12(4).* 295-311.

Sheridan, T., & Parasuraman, R. (2006). Human-automation interaction. *Reviews of Human Factors and Ergonomics*, 1, *89-129.* Retrieved 26 November, 2010 from http://archlab.gmu.edu/people/rparasur/Documents/Sheridan-ParasuramanRevHFES06.pdf

Suh, N. (2001). *Axiomatic design: Advances and applications.* NY: Oxford University Press

Using gold nanoparticles to hit cancer where it hurts. (2010). *Science Daily*. Retrieved 24 Feb 2012 from http://www.sciencedaily.com/releases/2010/02/10021610402.htm

Vicente, K. 1999. *Cognitive work analysis: Toward safe, productive and healthy computer based work*. NJ: L Erlbaum.

Woods, D., & Hollnagel, E (2006). *Joint cognitive systems: Patterns in cognitive systems engineering*. FL: CRC Press.

Woods, D.D., Patterson, E.S., & Roth, E.M.. (1998). Can we ever escape from data overload? A cognitive systems diagnosis. Cognitive Systems Engineering Laboratory Institute for Ergonomics: Ohio State University. Retrieved 19 April 2011, from http://10.1.1.37:6079

Appendix A: Informed Consent

CONSENT FORM

Embry-Riddle Aeronautical University

I voluntarily consent to participate in the research project entitled: **Measuring Mental Models of Human Factors and Software Engineers while Attempting to Solve the Same Problem – Effects of Complexity.** My participation will involve an introduction to and familiarization with a "mindmap" tool used for brainstorming called XMind, built on the Industry Standard Eclipse Foundation. This tool will be used by me in an individual setting to create a functional description of two prospective systems designs in the unmanned aircraft physical domain. This process will take between 90 minutes and 2 hours of my time.

The principal investigator of the study is: **Mr. Jerry Gordon.**

If I have questions about this study, I should inform Mr. Gordon or contact the following faculty member:

Dr. Elizabeth Blickensderfer
Human Factors and Systems Department
Embry-Riddle Aeronautical University
386-323-8065; blick488@my.erau.edu

I understand that the investigators believe that the risks or discomforts to me are as follows:

- *No greater than would be experienced during the performance of a typical in class homework assignment*
- *No greater than would be experienced when learning a simple new toolset in a professional setting*

The benefits that I may expect from my participation in this study are minimal. I understand that I will receive no direct benefit other than experience with a brainstorming tool and introduction to systems design in the unmanned aircraft problem domain.

My confidentiality during the study will be ensured by assigning me a coded identification number. My name will not be directly associated with any data. The confidentiality of the information related to my participation in this research will be ensured by maintaining records only coded by identification numbers.

The individual above, or their research assistants, have explained the purpose of the study, the procedures to be followed, and the expected duration of my participation. Possible benefits of the study have been described, as have alternative procedures, if such procedures are applicable and available.

I acknowledge that I have had the opportunity to obtain additional information regarding the study and that any questions I have raised have been answered to my full satisfaction.

Furthermore, I understand that I am free to withdraw consent at any time and to discontinue participation in the study without prejudice to me.

Finally, I acknowledge that I have read and fully understand the consent form. I sign it freely and voluntarily. A copy has been given to me.

Date: ____________________________

Name *(please print)*: ____________________
(Participant)

Signed: ____
(Participant)

Signed: ____
(Researcher/Assistant)

Appendix B: General Instructions

The purpose of this experiment is to measure the differences in the “mental models” or internal understanding of systems engineering and human factors students when approaching the same design problem. Based on the differences in their training and background, it is likely that they will conceptualize the same problem in different ways. Part of an individual’s “mental model” of a problem is a description of the **goals** and **behaviors** associated with the problem, as well as assigning means to accomplish those goals. This description is directly analogous to **functional requirements analysis**, and supports **functional allocation**, which is part of the engineering design process commonly used by both specialties.

Two problem statements will be provided, associated with new prospective designs for unmanned aircraft systems (UAS). The students will analyze the problem statements to develop a functional hierarchy for their proposed system design. Library resources and subject matter experts (SME) in the UAS domain can be used to provide background information or descriptions of existing systems. However, each student should complete the work without consulting their classmates, or a Cross-group professor (e.g. software students asking the Human Factors dept for help). At this time it is not required to propose any physical design solutions, i.e. to allocate functions to components, rather just to identify what the system is supposed to do and why.

The functional structure will be captured in a tool called XMind. This is a free download tool using an industry standard development environment called *Eclipse™*. Students pursuing careers in engineering are likely to experience a tool based on Eclipse at some point. This tool does not require registration, but participants are welcome to register with the site if they desire. A YouTube tutorial can be found here: http://www.youtube.com/watch?v=Ao5GakiCsqk

XMind is designed to facilitate brainstorming, without constraining the user to a particular process, or type of data. In this case, we want to focus on describing the **functions** of the **system** we are developing. Specifically, we will focus on developing a **functional flow** for the system under study. The ultimate result will start with a top level system **goal** that decomposes down to increasingly specific lower level goals and functions. For the purposes of this experiment, the "phases of flight" are given in both cases, the participant should decompose these phases into a series of goals, and a subordinate set of functions necessary to achieve those goals. XMind allows this to be done in either a top down (goals then functions), or a bottom up (functions then goals) fashion. Thus the final model should list the phases below a single common node, with two levels of decomposition below them created by each participant.

Functions/goals are described in a *present tense verb- direct object format* (e.g. store fuel, assign targets). Direct Object Nouns can be modified with adjectives if required (e.g. manage new tracks). Adverbs typically imply performance constraints (e.g. assign targets quickly), and are not appropriate. Likewise noun subjects, or indirect objects imply signaling or physical allocation (e.g. store fuel in the wings, weapon assigns targets) and are likewise inappropriate for describing functions.

However, describing **information flow** between functions is normally a part of the functional analysis process, described in **interface requirements.** For this experiment, identification of information flow is not required, only functions. However, describing interfaces might be useful, as all interfaces must have an originating and a receiving function and those are part of this exercise (however either the originator, or receiver, might be external to the system). From the above example, one function may be "assign targets" and another "prioritize targets" and the two functions might pass information like "target kinematic data" between them, and they may be allocated to different systems, such as a fire control system and a search system. While identifying "assign targets" and "prioritize targets" describe functions, the concept "target kinematic data" is information flow, and not required for this experiment.

If they aid in the User's analysis of the functional architecture, information flow can be captured in the tool. Using the XMind tool, relationships between concepts can be drawn, although they must be annotated as "information flow" or "allocation" if they are not functions. Most importantly, for this exercise, they need not be "complete" for the model, only as much as the student desires in performance of their analysis.

XMind allows for rearranging branches, so that ideas can free flow from the user. Topics can be identified in a strictly **top down** fashion, from lesser to greater levels of specificity, or in **bottom up** fashion, where the low level concepts are identified first and then grouped, once organizing concepts for the groups become more apparent. XMind allows for iterative creations where ideas can be grouped, regrouped, added, edited and deleted until the final hierarchy is complete. It is designed to present concepts graphically in a radial display. There is no specific "right answer" for choosing the organizing concepts – in fact this selection is an important part of the creative process. For the purposes of the experiment, at the end students should have between 15-30 low-level or "leaf node" concepts organized along with more than three levels of hierarchy.

XMind allows for two other useful concepts, a summary and a boundary. Summaries allow a comment to be made around a combination of sublevel functions Boundaries can be used to lasso around multiple function branches at multiple levels and used to specify what may be part of a system or subsystem, especially if the system has to relate to external systems.

Detailed Instructions

1. Download the XMind tool

The program can be downloaded from http://www.xmind.net/downloads/. It is free and you don’t have to register if you don’t want to. Once the installer is loaded on your computer, double click it to start. Select all options. When complete select “finished” and launch the tool from the start menu.

2. Start a new project or open your existing project, by clicking on file->open.

Name your file according to the project identifier on your demographic questionnaire – Mental_Model_xxx, where xxx is the number. Save the file as an XMInd workbook. An example is depicted in Figure 25.

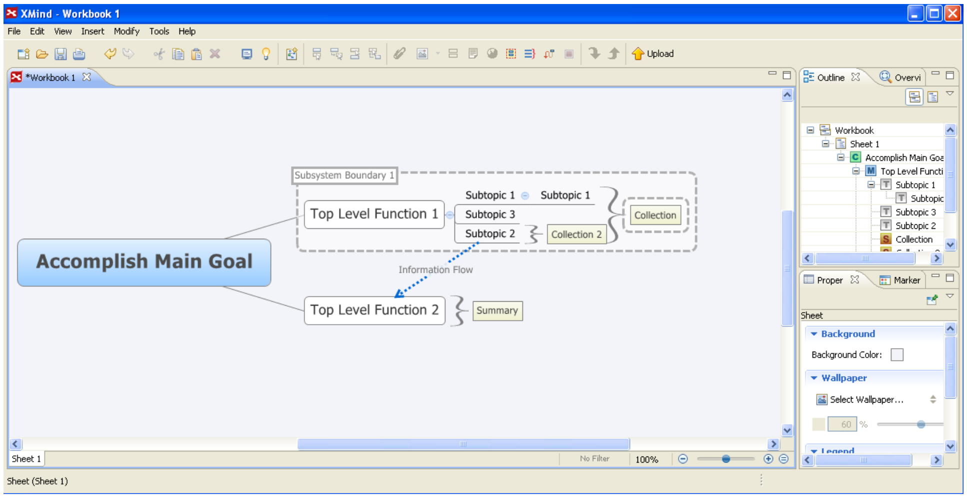

Figure 25. Example of the XMIND interface. Add Function icons are at the top. Functions appear in the large area to the left. Tree controls display the hierarchy in the upper right.

3. Start brainstorming your functional architecture

Start by defining a single high level goal for the system by clicking on “main topic” and replacing the text. Any concept can be updated at any time by double clicking and typing. Add terms as desired. The four icons in the center of the top frame (indicated in the red circle of Figure 20) enable you to add concepts at different levels, in order from left to right:

- Add function at the same level with the same parent
- Add a function at the next level down as a child of the highlighted function
- Add a function at the same level as the highlighted function, before this function in the list
- Add a parent function

4. Boundaries and Summaries can be added by using the icons within the green circle. Use Boundaries and Summaries to describe logical or system/subsystem boundaries. Relationships are added by clicking on the icon in the blue circle. Use relationships to specify information flows. Information flows should be nouns modified with adjectives. Again this is NOT REQUIRED for the experiment, unless it helps conceptualize the problem. Icons for these tasks are indicated in Figure 26.

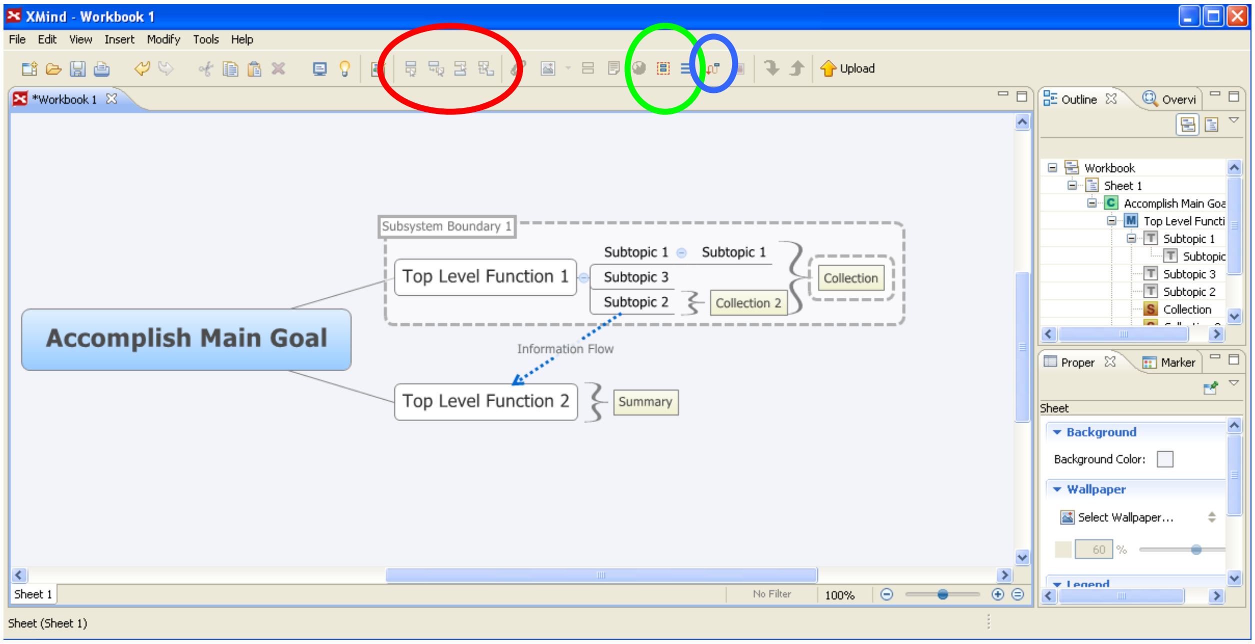

Figure 26. Icons used in Brainstorming. Not all are required to perform the experiment. Red indicates add Functions. Red brings up an editor menu and blue indicates relationship links.

5. Save you work and submit the final project file at the next class along with the demographic information.

XMind has many other features, but those are the essential ones necessary to conduct this experiment.

Appendix C: Demographic Questionnaire

Principal Language Spoken At home	English
	Other
Degree Program	HFS
	SY
Level	BS
	MS
Undergrad program	CS/EE/ME
	HF/Psych
	Other
	NA
Years industry experience	none
	<1
	1-5
	5+
I have experience Cross-group (HF-SW or SW-HF)	Y
	N
I have experience with using Visual Modeling tools (CORE, UML, etc)	Y
	N

Appendix D: Post Exercise Questionnaire

1. It is reasonable to expect that the open or “complex” problem was more challenging than the “closed” problem. However, they were design to be of roughly the same size in terms of functionality. How would you rate this “comparably sized” aspect of the problem statements on a scale of 1 – 5, where 1 is not the same and 5 is essentially the same size and scope (NOT difficulty).

2. Please provide a definition the following terms as they pertain to system design, and as you understand them from your coursework or textbooks:

Mission
Function
Task
System
Capability
Activity
Complexity

Appendix E: Closed System Problem Statement - Automated Ground Control Station

Part of the systems design process is to analyze requirements and identify a hierarchical representation of the functions which the system must perform. Each of these functions is ultimately “allocated” to physical components that must perform the work. It is these physical components which are purchased, developed or manufactured. The first system for which we are developing the “functional architecture” is a ground control station (GCS) for an unmanned aircraft system (UAS).

UAS include the robotic controlled aircraft, the Unmanned Aerial Vehicle (UAV) and the ground support personnel and equipment used to maintain and control the UAV. The system level behaviors of the entire UAS have an overall life cycle described as several “phases of flight”, similar to the life cycle of a manned aircraft. The basic “phases of flight” for the UAV are depicted in Figure 27.

Figure 27. Operation scenario or Phases of Flight for the Unmanned Aircraft System. Students are asked to focus on the green boxes.

The ground control system we are designing is only concerned with managing the functions associated with the phases and transitions depicted in green in Figure 2. Preflight and Post flight checks are conducted by a human ground crew. The aircraft is launched and landed by automated systems. Following launch, enroute activities include flying to (i.e. flyout) and return from the mission execution area (25 KM range). Mission execution involves operating the payload as needed, and monitoring the aircraft and environment for mechanical failures, inclement weather and hostile fire. Recovery landing is initiated by the operator during enroute return when the aircraft has achieved the recovery window altitude, heading and speed. Overall context diagrams for the UAS, including the green GCS segment of which we are concerned in depicted in Figure 28.

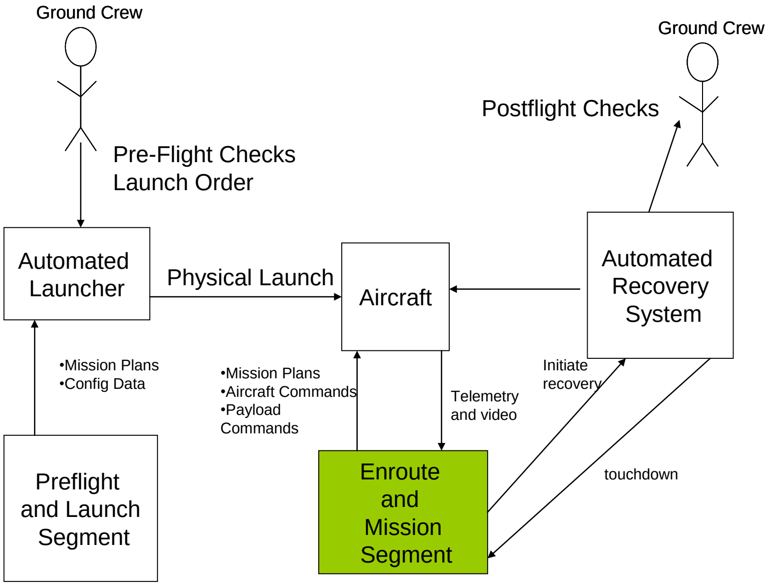

Figure 28. Context Diagram of Ground Control Station (GCS). The Functions being developed by participants will operate within the subsystem/segment indicated in green.

The aircraft managed by the GCS is a medium range unmanned aircraft, capable of autonomous flight. It is equipped with an optical and infrared camera payload (no weapons). The camera has a “point at location” and motion tracking feature, and can be steered manually by the ground operator. It is used to conduct intelligence, surveillance and reconnaissance (ISR) missions for the US Army. It is powered by a single “pusher” propeller engine.

The aircraft has an onboard autopilot which receive and upload and execute automated mission plans from the GCS, (AUTONAV mode) or keep a specific constant heading, altitude and speed as directed by the ground operator (Knobs Mode). Mission plans can be revised by the ground station at any time. The aircraft payload and autopilot can be transferred and controlled individually or together from another GCS, and/or returned to the master GCS.

Aircraft systems that are monitored and controlled using the GCS include:

- Engine (with temp sensors)
- Power Distribution (with voltage and amperage sensor)
- Autopilot
- Pitot/static system
- GPS
- Radio Links (payload and aircraft control uplink and downlink)
- Payload (IR and Visual Camera, 0-200x zoom)
- Control Surface Actuators
- Identify Friend or Foe(IFF)
- A parachute emergency recovery system

On the current system, ground control is performed by two human operators – one to work the aircraft and one to work the payload. The army desires to use automation to reduce this to one combined operator. The first part of this effort will be to describe the functions which must be performed. Once the functions are identified, they are allocated to components, including hardware and software automation. Your task is to describe the functional architecture for the ground control system that can manage these phases:

Flyout Enroute – movement from the launch and recovery area to the objective area

Mission execution,- execution of ISR mission as requested by other units

Return Enroute- movement from the objective area to the launch and recovery area

Appendix F – Open System Problem Statement - Automated Mission Control System

The second system for which we are developing the "functional architecture" is an automated mission control system (AMCS) for the UAS described in the first problem. This system can be installed in ground vehicles, static ground sites, shipboard, or aboard other aircraft. One or more of the systems may work simultaneously in conjunction with one or more other systems, controlling and interfacing with multiple aircraft and supported units. The system will work with the automated GCS and the other automated components of the UAS, as described in the first problem, to provide interface between the aircraft, higher headquarters (HHQ) and the various combat units that might request ISR support from a UAS: infantry commanders, Attack Aircraft conducting Close Air Support (CAS), Engineers/Route Clearance, Special Operations, Combat Search and Rescue (CSAR), and Support Fires (Artillery and Naval Gunfire) etc.

The Mission Life Cycle of this system is depicted in Figure 29.

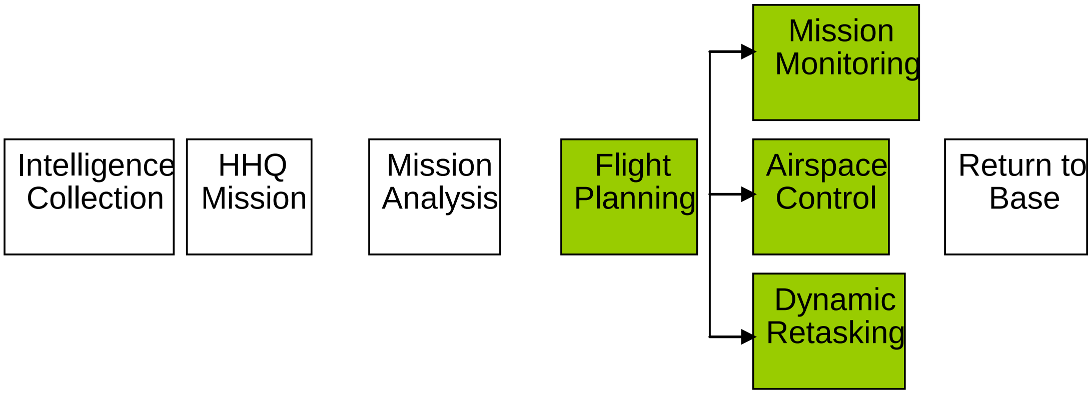

Figure 29. Phases of the UAS Mission Life Cycle. Multiple individual aircraft phases of flight may be executed within the 3 vertical green boxes by multiple aircraft Participants are asked to develop functional flow for the green boxes.

This new system will support capabilities of Tactical Operations Centers (TOC) and Air Support Operations Centers (ASOC), currently requiring a number of dedicated operators to perform manual tasks. The TOC manages mission assignments and the ASOC tracks military aircraft and fuel/ammunition/facility resources, The Army wishes to use extensive automation in this system, while pushing the remaining task work out to other already existing operators (e.g. the UAS ground operator, vehicle operators, ATC) by participation in an overall command and control network (i.e. a net personnel reduction).

The system we are designing focuses on the green boxes of Figure 4. HHQ staff officers are the ultimate users of the system; however it may rely on a number of other operators. It will interface directly with the aircraft Ground Control Station (GCS) through an automated mission planning capability, available when the aircraft is in “AUTONAV” mode. It will also enable and coordinate the control hand off of the payload and/or the autopilot. It can utilize data and voice communications (to include chat/IM functions) as necessary to communicate between the UAS operators at the GCSs, the staff operators at ASOC, TOC, ATC sites, and any other vehicle operators which are part of its command and control network. It must manage the multiple aircraft/multiple unit/multiple installation problem (figuring out what “management” means is a critical portion of the functional analysis). The overall system context is depicted in Figure 30.

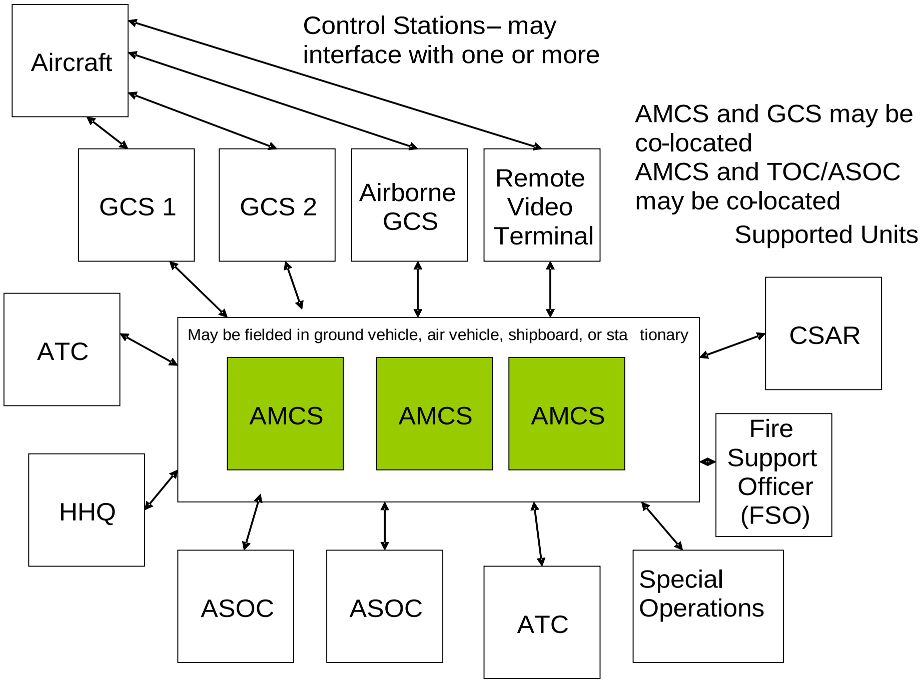

Figure 30. Context Diagram for the AMCS. This shows external interfaces which must collaborate with the AMCS, and for which the AMCS might pick up some additional functions. AMCS components addressed by participants are in the green boxes, which may be fielded in multiple systems simultaneously.

Flight Planning includes the information normally found in HHQ plans, along with additional detail for the unit to plan flight sorties. This includes:

- Tasked unit
- Takeoff and landing locations
- Alternate Landing Sites
- Fuel Loading
- Callsigns
- Transponder codes
- Air Traffic Control Frequencies
- Assigned Missions

- Tail/Side number of the aircraft used
- High Value Targets (HVT)
- Targets area of Interest (TAI)
- Named Areas of Interest (NAI)
- Supported Units

Mission Monitoring includes the activities associated with sharing or interpreting ISR imagery from the UAS camera payload, and any other things necessary for the staffs to maintain situation awareness of the vehicle's ability to complete current and future missions. Monitoring may be done locally at the GCS, in the field by a unit using a remote video terminal (RVT – one way repeater of the payload picture) or by the AMCS directly.

Airspace control includes those system functions necessary to operate within the Air Traffic Control (ATC) procedures and systems in place where the UAS is flying. Note this system doesn't replace ATC, it interfaces with existing ATC assets and provides some of their duties for the UAS.

Dynamic retasking includes the ability to manage multiple requests from different units which might require UAS ISR support, and provide changes to the planned mission based on those requests in real time as the mission is flown. Retasking might address transfer of controls as necessary to support.

This task is to analyze the behaviors of the system as explained above. In order to accomplish this, describe a hierarchical functional architecture for the Automated Mission control system that can manage flight planning, mission monitoring, airspace control and dynamic retasking of the UAS. It must support multiple fielding configurations and interface with all varieties of supported units and provide coordination with ATC and Higher Commands such as HHQ, ASOC and TOC. A list of acronyms that apply to the open and closed system problems is provided in Table 15.

Table 15
List of acronyms used within the problem cases

Acronym	Full Name	Definition
AMCS	Automated Mission Control System	Our Fictional Complex System
ASOC	Air Support Operations Center	Manages Aircraft for operational staff in large scale battlefields
ATC	Air Traffic Control	Performs the same functions as domestic ATC
AUTONAV	Auto navigation	Autopilots executing Fly to points automatically
CAS	Close Air Support	Friendly aircraft attacking enemy ground targets located near friendly forces
CSAR	Combat Search and Rescue	Recovery of a lost soldier/sailor/airman under combat conditions
FSO	Fire Support Officer	Coordinates use of Artillery and Naval Gun Fire with ground forces
GCS	Ground Control Station	Where UAV operators control aircraft remotely
GPS	Global Positioning System	A satellite based navigation aid
HHQ	Higher Head Quarters	The admiral or general commanding a force
HVT	High Value Target	An enemy which is critical to the battle
IFF	Identify Friend or Foe	An electronic system for identifying aircraft
IR	Infrared	Cameras which detect outside the visual spectrum
ISR	Intelligence, Surveillance and Reconnaissance	Mission area which employs UAS as eyes in the sky
NAI	Named Area of interest	An area which must be observed for enemy intentions
RVT	Remote Video Terminal	An monitor which can tune into a UAV camera, not located with GCS
TAI	Target Area of Interest	An area of the battlefield which is to be attacked, held or cordoned
TOC	Tactical Operations Center	The lowest level of command and control which coordinate UAV missions
UAS	Unmanned Aircraft System	The unmanned aircraft and its ground based support systems
UAV	Unmanned Aerial Vehicle	A small highly automated aircraft which is directed or piloted remotely from the ground.

Appendix G: Raw Data Parent Concepts

Group	Participant	Problem Type	Phase	Goal/Function
SY	7	open	Mission Monitoring	Distribute ISR imagery
SY	7	open	Mission Monitoring	Distribute vehicle status
SY	7	open	airspace control	interface with ATC
SY	7	open	airspace control	deconflict mission flight plans
SY	7	open	dynamic retasking	manage ISR support requests
SY	7	open	dynamic retasking	transfer controls
SY	7	open	flight planning	manage UAS mission plans
SY	7	open	flight planning	disseminate flight plans
SY	7	closed	enroute fly out	monitor flight
SY	7	closed	enroute fly out	control flight
SY	7	closed	enroute fly out	assume control of aircraft
SY	7	closed	enroute return	monitor flight
SY	7	closed	enroute return	control flight
SY	7	closed	enroute return	prepare for landing
SY	7	closed	execute mission	monitor flight
SY	7	closed	execute mission	control flight
SY	7	closed	execute mission	survey target area
SY	13	closed	enroute return	check systems
SY	13	closed	enroute return	conduct communications
SY	13	closed	enroute fly out	control aircraft
SY	13	closed	enroute fly out	conduct communications
SY	13	closed	execute mission	operate payload
SY	13	closed	execute mission	perform ISR
SY	13	open	dynamic retasking	manage resupply
SY	13	open	dynamic retasking	manage air combat
SY	13	open	airspace control	manage air traffic
SY	13	open	airspace control	support land component
SY	13	open	flight planning	Calculate route
SY	13	open	flight planning	determine flight distance
SY	13	open	Mission Monitoring	monitor payload
SY	13	open	Mission Monitoring	monitor flight
SY	13	open	Mission Monitoring	monitor payload
SY	18	open	Mission Monitoring	Distribute ISR imagery
SY	18	open	Mission Monitoring	maintain SA
SY	18	open	manage airspace control	interface with ATC
SY	19	open	manage airspace control	interface with supported units

Group	Participant	Problem Type	Phase	Goal/Function
SY	18	open	dynamic retasking	manage ISR support requests
SY	18	open	dynamic retasking	provide changes to mission
SY	18	open	dynamic retasking	transfer controls
SY	18	open	flight planning	Determine Tail/Side Number of the Aircraft Used
SY	18	open	flight planning	Manage Supported Units
SY	18	open	flight planning	Manage Call Signs
SY	18	open	flight planning	Assign High Value Targets (HVT)
SY	18	open	flight planning	Determine Alternate Landing Sites
SY	18	open	flight planning	Manage Assigned Missions
SY	18	open	flight planning	Determine Target Areas of Interest (TAI)
SY	18	open	flight planning	Coordinate with Tasked Unit
SY	18	open	flight planning	Determine Named Areas of Interest (NAI)
SY	18	open	flight planning	Manage Fuel Loading
SY	18	open	flight planning	Determine Takeoff and Landing Locations
SY	18	open	flight planning	Manage Air Traffic Control Frequencies
SY	18	open	flight planning	Manage Transponder Codes
SY	18	closed	enroute return	prepare for landing
SY	18	closed	enroute return	monitor flight
SY	18	closed	enroute return	control flight
SY	18	closed	enroute fly out	monitor flight
SY	18	closed	enroute fly out	control flight
SY	18	closed	enroute fly out	proceed to mission area
SY	37	closed	enroute fly out	proceed to mission area
SY	37	closed	execute mission	operate payload
SY	37	closed	execute mission	monitor flight
SY	37	closed	enroute return	prepare for landing
SY	37	open	dynamic retasking	manage ISR support requests
SY	37	open	Mission Monitoring	Distribute ISR imagery
SY	37	open	Mission Monitoring	maintain SA
SY	37	open	airspace control	interface with ASOC
SY	4	closed	enroute return	Obtain current location information
SY	4	closed	enroute return	Obtain target destination information
SY	4	closed	enroute return	Calculate route
SY	4	closed	enroute return	monitor flight
SY	4	closed	enroute return	control flight
SY	4	closed	execute mission	maintain SA
SY	4	closed	execute mission	back up mission data

Group	Participant	Problem Type	Phase	Goal/Function
SY	4	open	Mission Monitoring	obtain comm link
SY	4	open	Mission Monitoring	verify comm link
SY	4	open	Mission Monitoring	verify comm security
SY	4	open	Mission Monitoring	command data
SY	4	open	Mission Monitoring	receive data
SY	4	open	Mission Monitoring	verify data
SY	4	open	Mission Monitoring	process data
SY	4	open	dynamic retasking	obtain tasked unit
SY	4	open	dynamic retasking	manage facility
SY	4	open	dynamic retasking	Manage Fuel Loading
SY	4	open	dynamic retasking	Manage Call Signs
SY	4	open	dynamic retasking	Manage Transponder Codes
SY	4	open	dynamic retasking	manage ATC
SY	4	open	dynamic retasking	receive mission and targets
SY	4	open	dynamic retasking	receive tail number
SY	4	open	dynamic retasking	Determine Target Areas of Interest (TAI)
SY	4	open	dynamic retasking	obtain HVT
SY	4	open	dynamic retasking	obtain supported units
SY	4	open	airspace control	evaluate HHQ priority
SY	4	open	airspace control	interface with ASOC
SY	4	open	airspace control	interface with airborne GCS
SY	4	open	airspace control	handle emergencies
SY	4	open	airspace control	manage ATC
SY	4	open	airspace control	manage fire support
SY	4	open	airspace control	manage CSAR
SY	4	open	airspace control	manage SPECOPS
SY	4	open	airspace control	monitor RVT
SY	12	closed	enroute fly out	proceed to mission area
SY	12	closed	enroute fly out	receive mission and targets
SY	12	closed	execute mission	operate payload
SY	12	closed	execute mission	monitor flight
SY	12	closed	enroute return	Obtain target destination information
SY	12	closed	enroute return	control flight
SY	12	open	dynamic retasking	manage ISR support requests
SY	12	open	dynamic retasking	evaluate HHQ priority
SY	12	open	dynamic retasking	conduct communications
SY	12	open	Mission Monitoring	conduct communications
SY	12	open	Mission Monitoring	Distribute ISR imagery
SY	12	open	flight planning	disseminate flight plans
SY	12	open	flight planning	update flight plans

Group	Participant	Problem Type	Phase	Goal/Function
SY	12	open	flight planning	conduct communications
SY	11	closed	enroute fly out	assume control of aircraft
SY	11	closed	execute mission	monitor flight
SY	11	closed	execute mission	control flight
SY	11	closed	enroute return	prepare for landing
SY	11	closed	Mission monitoring	Support TOC
SY	11	closed	Mission monitoring	interface with ASOC
SY	11	closed	airspace control	conduct communications
SY	11	closed	airspace control	manage facility
SY	11	closed	airspace control	manage aircraft
SY	11	closed	dynamic retasking	manage ISR support requests
SY	2	closed	enroute return	navigate aircraft
SY	2	closed	enroute return	conduct communications
SY	2	closed	enroute fly out	navigate aircraft
SY	2	closed	enroute fly out	conduct communications
SY	2	closed	execute mission	navigate aircraft
SY	2	closed	execute mission	conduct communications
SY	2	closed	execute mission	operate payload
SY	2	open	Mission Monitoring	manage imagery
SY	2	open	Mission Monitoring	monitor flight
SY	2	open	airspace control	interface with ATC
SY	2	open	airspace control	navigate aircraft
SY	2	open	dynamic retasking	manage ISR support requests
SY	2	open	dynamic retasking	transfer controls
SY	14	closed	enroute fly out	control Autonav mode
SY	14	closed	enroute fly out	control knobs mode
SY	14	closed	enroute return	control Autonav mode
SY	14	closed	enroute return	control knobs mode
SY	14	closed	execute mission	operate payload
SY	14	open	dynamic retasking	provide changes to mission
SY	14	open	Mission Monitoring	monitor flight
SY	14	open	Mission Monitoring	monitor mission navigation
SY	14	open	airspace control	track aircraft
SY	14	open	enroute fly out	control Autonav mode
SY	14	open	enroute fly out	control knobs mode
SY	14	open	enroute fly out	monitor flight
SY	14	open	enroute return	control Autonav mode
SY	14	open	enroute return	control knobs mode
SY	14	open	enroute return	monitor flight
SY	14	open	execute mission	operate payload
SY	14	open	execute mission	monitor flight

Group	Participant	Problem Type	Phase	Goal/Function
SY	16	open	flight planning	Assign Take off and Landing Locations
SY	16	open	flight planning	Assign Alternate Landing Sites
SY	16	open	flight planning	Assign Callsign
SY	16	open	flight planning	Assign Transponders Codes
SY	16	open	flight planning	Determine Fuel Loading
SY	16	open	flight planning	Assign ATC Freq
SY	16	open	flight planning	Assign TAI
SY	16	open	flight planning	Assign NAI
SY	16	open	flight planning	conduct communications
SY	16	open	dynamic retasking	manage ISR support requests
SY	16	open	Mission Monitoring	manage imagery
SY	16	open	Mission Monitoring	monitor flight
SY	16	open	Mission Monitoring	conduct communications
SY	16	closed	enroute fly out	control Autonav mode
SY	16	closed	enroute fly out	control knobs mode
SY	16	closed	enroute fly out	transfer controls
SY	16	closed	enroute fly out	monitor flight
SY	16	closed	enroute return	control Autonav mode
SY	16	closed	enroute return	control knobs mode
SY	16	closed	enroute return	transfer controls
SY	16	closed	enroute return	monitor flight
SY	16	closed	execute mission	operate payload
SY	16	closed	execute mission	monitor flight
SY	16	closed	execute mission	transfer controls
SY	17	closed	enroute fly out	control speed
SY	17	closed	enroute fly out	control altitude
SY	17	closed	enroute return	control speed
SY	17	closed	enroute return	control altitude
SY	17	closed	execute mission	monitor flight
SY	17	open	dynamic retasking	evaluate HHQ priority
SY	17	open	dynamic retasking	identify targets
SY	17	open	airspace control	manage ISR support requests
SY	17	open	airspace control	interface with ATC
SY	17	open	airspace control	interface with supported units
SY	17	open	airspace control	control Autonav mode
SY	17	open	Mission Monitoring	Assign Take off and Landing Locations
SY	17	open	Mission Monitoring	receive mission and targets
SY	17	open	Mission Monitoring	manage imagery
SY	19	closed	enroute fly out	monitor flight

Group	Participant	Problem Type	Phase	Goal/Function
SY	19	closed	enroute fly out	interact with users
SY	19	closed	enroute fly out	control altitude
SY	19	closed	enroute fly out	control flight
SY	19	closed	mission execution	monitor flight
SY	19	closed	mission execution	interact with users
SY	19	closed	mission execution	control flight
SY	19	closed	enroute return	prepare for landing
SY	19	closed	enroute return	interact with users
SY	19	closed	enroute return	control flight
SY	19	open	dynamic retasking	handle emergencies
SY	19	open	dynamic retasking	develop flight plans
SY	19	open	dynamic retasking	interact with users
SY	19	open	mission monitoring	maintain flight
SY	19	open	mission monitoring	interact with users
SY	19	open	mission monitoring	monitor flight
SY	19	open	airspace control	maintain flight
SY	19	open	airspace control	identify airspace class
SY	19	open	airspace control	interact with users
HF	1	closed	enroute fly out	conduct pre-flight checks
HF	1	closed	enroute fly out	obtain clearances
HF	1	closed	enroute fly out	launch
HF	1	closed	execute mission	operate payload
HF	1	closed	execute mission	monitor aircraft
HF	1	closed	execute mission	monitor environment
HF	1	closed	execute mission	monitor systems
HF	1	closed	execute mission	conduct mission
HF	1	closed	execute mission	fly to waypoint
HF	1	closed	enroute return	initiate return
HF	1	closed	enroute return	obtain clearances
HF	1	closed	enroute return	conduct post flight checks
HF	1	closed	enroute return	land
HF	1	open	flight planning	configure mission plan
HF	1	open	flight planning	configure flight plan
HF	1	open	mission monitoring	coordinate with ASOC
HF	1	open	mission monitoring	coordinate with TOC
HF	1	open	mission monitoring	interpret imagery
HF	1	open	mission monitoring	maintain SA
HF	1	open	airspace control	operate with ATC
HF	1	open	dynamic retasking	Process ISR request
HF	1	open	dynamic retasking	provide changes to mission
HF	8	closed	enroute return	conduct communications

Group	Participant	Problem Type	Phase	Goal/Function
HF	8	closed	enroute return	conduct communications
HF	8	closed	execute mission	operate payload
HF	8	closed	execute mission	monitor aircraft
HF	8	closed	execute mission	conduct communications
HF	8	closed	execute mission	monitor flight
HF	3	closed	enroute fly out	monitor systems
HF	3	closed	enroute fly out	control systems
HF	3	closed	execute mission	monitor systems
HF	3	closed	execute mission	control systems
HF	3	closed	enroute return	monitor systems
HF	3	closed	enroute return	control systems
HF	3	open	dynamic retasking	manage ISR support requests
HF	3	open	dynamic retasking	provide changes to mission
HF	3	open	dynamic retasking	transfer controls
HF	3	open	airspace control	interface with supported units
HF	3	open	mission monitoring	mission monitoring
HF	3	open	mission monitoring	interpret imagery
HF	3	open	mission monitoring	maintain SA
HF	3	open	mission monitoring	manage imagery
HF	3	open	flight planning	Assign Alternate Landing Sites
HF	3	open	flight planning	Assign Callsign
HF	3	open	flight planning	Assign Transponders Codes
HF	3	open	flight planning	Determine Fuel Loading
HF	3	open	flight planning	Assign ATC Freq
HF	3	open	flight planning	Assign TAI
HF	3	open	flight planning	Assign NAI
HF	3	open	flight planning	Coordinate with Tasked Unit
HF	3	open	flight planning	Determine Takeoff and Landing Locations
HF	3	open	flight planning	Assign High Value Targets (HVT)
HF	3	open	flight planning	Determine Tail/Side Number of the Aircraft Used
HF	3	open	flight planning	Assign High Value Targets (HVT)
HF	3	open	flight planning	Manage Supported Units
HF	3	open	flight planning	assign task unit
HF	5	closed	enroute fly out	proceed to waypoint
HF	5	closed	enroute fly out	monitor airspace
HF	5	closed	enroute fly out	monitor communications
HF	5	closed	enroute fly out	monitor fuel
HF	5	closed	enroute fly out	monitor weapons
HF	5	closed	enroute fly out	monitor weather

Group	Participant	Problem Type	Phase	Goal/Function
HF	5	closed	enroute fly out	execute lost link procedure
HF	5	closed	execute mission	process CAS
HF	5	closed	execute mission	process CSAR request
HF	5	closed	execute mission	process SEAD request
HF	5	closed	execute mission	Process ISR request
HF	5	closed	execute mission	Provide imagery
HF	5	open	enroute fly out	proceed to waypoint
HF	5	open	enroute fly out	monitor airspace
HF	5	open	enroute fly out	monitor communications
HF	5	open	enroute fly out	monitor fuel
HF	5	open	enroute fly out	monitor weapons
HF	5	open	enroute fly out	monitor weather
HF	5	open	enroute fly out	execute lost link procedure
HF	5	open	mission monitoring	process CAS
HF	5	open	mission monitoring	Provide imagery
HF	5	open	dynamic retasking	process CSAR request
HF	5	open	dynamic retasking	process SEAD request
HF	5	open	dynamic retasking	Process ISR request
HF	5	open	enroute fly out	proceed to waypoint
HF	5	open	enroute fly out	monitor aircraft
HF	5	open	enroute fly out	monitor environment
HF	9	closed	execute mission	operate payload
HF	9	closed	execute mission	control aircraft
HF	9	closed	execute mission	control payload
HF	9	closed	execute mission	monitor environment
HF	9	closed	execute mission	monitor aircraft
HF	9	closed	enroute return	proceed to waypoint
HF	9	closed	enroute return	monitor aircraft
HF	9	closed	enroute return	monitor environment
HF	9	open	mission monitoring	support TOC
HF	9	open	mission monitoring	support ASOC
HF	9	open	mission monitoring	interpret information
HF	9	open	control airspace	receive communication
HF	9	open	control airspace	manage aircraft
HF	9	open	dynamic retasking	monitor requests
HF	9	open	dynamic retasking	manage requests
HF	15	closed	enroute fly out	monitor engine
HF	15	closed	enroute fly out	monitor GPS
HF	15	closed	enroute fly out	monitor power
HF	15	closed	enroute fly out	monitor instruments
HF	15	closed	enroute fly out	monitor heading

Group	Participant	Problem Type	Phase	Goal/Function
HF	15	closed	enroute fly out	control surfaces
HF	15	closed	execute mission	Process ISR request
HF	15	closed	enroute return	monitor engine
HF	15	closed	enroute return	monitor GPS
HF	15	closed	enroute return	control parachute
HF	15	open	flight planning	configure mission plan
HF	15	open	flight planning	manage flight plan
HF	15	open	mission monitoring	interface with ASOC
HF	15	open	mission monitoring	interface with CSAR
HF	15	open	mission monitoring	interface with TOC
HF	15	open	airspace control	interface with ATC
HF	15	open	dynamic retasking	transfer controls
HF	15	open	dynamic retasking	manage requests
HF	15	open	dynamic retasking	update mission
HF	20	closed	enroute return	update mission
HF	20	closed	enroute return	navigate aircraft
HF	20	closed	enroute return	enter recovery parameters
HF	20	closed	enroute fly out	update mission
HF	20	closed	enroute fly out	navigate aircraft
HF	20	closed	enroute fly out	control surfaces
HF	20	closed	execute mission	select payload
HF	20	closed	execute mission	control payload
HF	20	open	dynamic retasking	manage requests
HF	20	open	dynamic retasking	update mission
HF	20	open	mission monitoring	select operator
HF	20	open	mission monitoring	manage imagery
HF	20	open	mission monitoring	monitor aircraft
HF	20	open	mission monitoring	receive mission and targets
HF	20	open	control airspace	define airspace
HF	20	open	control airspace	communicate with airspace

Appendix H: Raw Data Child Concepts

Group	Participant	Problem Type	Sub-function
SY	7	closed	request clearance for landing
SY	7	closed	initiate landing
SY	7	closed	notify ground
SY	7	closed	release control
SY	7	closed	control zoom
SY	7	closed	control payload mode
SY	7	closed	control target
SY	7	closed	monitor video
SY	7	closed	set pattern
SY	7	closed	store mission plan
SY	7	closed	update mission plan
SY	7	closed	control mode change
SY	7	closed	control aircraft
SY	7	closed	engage parachute
SY	7	closed	monitor location
SY	7	closed	monitor alerts
SY	7	closed	monitor engine
SY	7	closed	transfer control
SY	7	closed	transfer payload
SY	7	closed	receive control
SY	7	closed	receive payload
SY	7	closed	maintain communication
SY	7	closed	initialize mission plan
SY	7	open	maintain airfields
SY	7	open	manage fuel load
SY	7	open	manage ISR targets
SY	7	open	disseminate flight plans
SY	7	open	notify ATC
SY	7	open	notify retasking module
SY	7	open	manage ISR requests
SY	7	open	evaluate ISR requests
SY	7	open	Assign vehicle
SY	7	open	update mission plan
SY	7	open	notify transfer
SY	7	open	transfer control
SY	7	open	manage ISR requests
SY	7	open	evaluate ISR requests
SY	7	open	transmit video
SY	7	open	monitor systems
SY	7	open	disseminate system state
SY	7	open	manage ATC frequencies

Group	Participant	Problem Type	Sub-function
SY	7	open	manage tail number
SY	7	open	manage call sign
SY	7	open	manage flight plans
SY	7	open	evaluate flight plans
SY	7	open	modify flight plans
SY	7	open	disseminate flight plans
SY	13	closed	check payload
SY	13	closed	check control system
SY	13	closed	communicate with GPS
SY	13	closed	track radar
SY	13	closed	control autopilot
SY	13	closed	control surfaces
SY	13	closed	navigate aircraft
SY	13	closed	maintain communication
SY	13	closed	locate enemy
SY	13	closed	identify friendlies
SY	13	closed	relay information
SY	13	closed	monitor cargo
SY	13	closed	monitor instruments
SY	13	closed	monitor ammunition
SY	13	closed	call for fire
SY	13	open	conduct refueling
SY	13	open	transfer cargo
SY	13	open	conduct search
SY	13	open	Conduct rescue
SY	13	open	identify landing site
SY	13	open	avoid collision
SY	13	open	conduct fire support
SY	13	open	remote control aircraft
SY	13	open	identify takeoff site
SY	13	open	identify alternate landing sites
SY	13	open	calculate flight time
SY	13	open	determine terrain
SY	13	open	calculate fuel load
SY	13	open	calculate nav methods
SY	13	open	transfer imagery
SY	13	open	monitor cargo status
SY	13	open	monitor environment
SY	13	open	display location
SY	18	open	maintain GCS status
SY	18	open	maintain RVT status
SY	18	open	support configurations
SY	18	open	maintain AMCS status
SY	18	open	monitor fuel

Group	Participant	Problem Type	Sub-function
SY	18	open	monitor ammunition
SY	18	open	maintain communication
SY	18	open	track aircraft
SY	18	open	manage assignments
SY	18	open	interface with ASOC
SY	18	open	interface with TOC
SY	18	open	interface with ATC
SY	18	open	track tail number
SY	18	open	manage supported units
SY	18	open	manage call signs
SY	18	open	assign HVT
SY	18	open	identify alternate landing sites
SY	18	open	monitor mission
SY	18	open	assign TAI
SY	18	open	assign NAI
SY	18	open	coordinate with tasked unit
SY	18	open	calculate fuel load
SY	18	open	identify takeoff site
SY	18	open	identify landing site
SY	18	open	manage transponder
SY	18	open	manage ATC frequencies
SY	18	open	manage ISR requests
SY	18	open	disseminate
SY	18	open	transfer control
SY	18	open	relay mission change
SY	18	open	control mode change
SY	18	open	interface with GCS
SY	18	closed	monitor engine
SY	18	closed	monitor power
SY	18	closed	monitor autopilot
SY	18	closed	monitor instruments
SY	18	closed	monitor GPS
SY	18	closed	monitor payload
SY	18	closed	monitor surfaces
SY	18	closed	manage transponder
SY	18	closed	monitor parachute
SY	18	closed	maintain communication
SY	18	closed	control power
SY	18	closed	control autopilot
SY	18	closed	control instruments
SY	18	closed	control GPS
SY	18	closed	control payload
SY	18	closed	control surfaces
SY	18	closed	control transponder

Group	Participant	Problem Type	Sub-function
SY	18	closed	control parachute
SY	37	closed	control flight
SY	37	closed	manage range
SY	37	closed	operate payload
SY	37	closed	monitor engine
SY	37	closed	monitor power
SY	37	closed	monitor autopilot
SY	37	closed	monitor instruments
SY	37	closed	monitor GPS
SY	37	closed	maintain communication
SY	37	closed	control payload mode
SY	37	closed	control surfaces
SY	37	closed	manage transponder
SY	37	closed	prepare for landing
SY	37	closed	control parachute
SY	37	open	manage ISR requests
SY	37	open	assign HVT
SY	37	open	assign TAI
SY	37	open	manage CSAR
SY	37	open	interface with ATC
SY	37	open	interface with GCS
SY	37	open	distribute ISR to GCS
SY	37	open	distribute ISR to field unit
SY	37	open	distribute ISR to AMCS
SY	37	open	maintain SA
SY	37	open	monitor GPS
SY	4	open	send security data
SY	4	open	verify link strength
SY	4	open	maintain communication
SY	4	open	send data
SY	4	open	receive data
SY	4	open	verify data
SY	4	open	process data
SY	4	open	manage supported units
SY	4	open	identify features
SY	4	open	locate units
SY	4	open	describe units
SY	4	open	identify alternate landing sites
SY	4	open	identify takeoff site
SY	4	open	identify landing site
SY	4	open	calculate fuel load
SY	4	open	manage call sign
SY	4	open	manage transponder
SY	4	open	obtain missions

Group	Participant	Problem Type	Sub-function
SY	4	open	manage tail number
SY	4	open	assign HVT
SY	4	open	assign TAI
SY	4	open	assign NAI
SY	4	open	interface with HHQ
SY	4	open	interface with ASOC
SY	4	open	interface with GCS
SY	4	open	interface with ATC
SY	4	open	relay locations
SY	4	open	relay flight plan
SY	4	open	disseminate flight plans
SY	4	open	conduct fire support
SY	4	open	manage CSAR
SY	4	open	manage SPECOPS
SY	4	open	manage RVT
SY	4	closed	update mission plan
SY	4	closed	monitor GPS
SY	4	closed	obtain obstacle data
SY	4	closed	get location
SY	4	closed	get destination
SY	4	closed	calculate flight time
SY	4	closed	monitor engine
SY	4	closed	monitor power
SY	4	closed	monitor autopilot
SY	4	closed	monitor instruments
SY	4	closed	monitor GPS
SY	4	closed	monitor communications
SY	4	closed	control surfaces
SY	4	closed	monitor transponder
SY	4	closed	monitor parachute
SY	4	closed	alert emergencies
SY	4	closed	control heading
SY	4	closed	control speed
SY	4	closed	calculate thrust
SY	4	closed	calculate control inputs
SY	4	closed	sense obstacles
SY	4	closed	update route
SY	4	closed	store mission plan
SY	4	closed	store imagery
SY	4	closed	store health
SY	4	closed	obtain missions
SY	4	closed	evaluate mission
SY	4	closed	update route
SY	4	closed	compare imagery

Group	Participant	Problem Type	Sub-function
SY	12	closed	obtain missions
SY	12	closed	receive targets
SY	12	closed	proceed to mission area
SY	12	closed	operate payload camera
SY	12	closed	point at target
SY	12	closed	track target
SY	12	closed	monitor engine
SY	12	closed	monitor environment
SY	12	closed	monitor power
SY	12	closed	monitor autopilot
SY	12	closed	monitor instruments
SY	12	closed	monitor GPS
SY	12	closed	monitor communications
SY	12	closed	monitor payload
SY	12	closed	monitor surfaces
SY	12	closed	monitor transponder
SY	12	closed	control parachute
SY	12	closed	obtain target
SY	12	closed	control flight
SY	12	open	communicate with HHQ
SY	12	open	communicate with GCS
SY	12	open	communicate with supported units
SY	12	open	evaluate target priorities
SY	12	open	manage ISR requests
SY	12	open	monitor ATC
SY	12	open	distribute ISR to message server
SY	12	open	distribute ISR to RVT
SY	12	open	display location
SY	12	open	disseminate tasked units
SY	12	open	identify alternate landing sites
SY	12	open	manage fuel load
SY	12	open	manage callsigns
SY	12	open	manage transponder
SY	12	open	manage ATC frequencies
SY	12	open	assign HVT
SY	12	open	assign TAI
SY	12	open	assign NAI
SY	12	open	manage supported units
SY	11	closed	control launch
SY	11	closed	guide launch
SY	11	closed	calculate max range
SY	11	closed	control heading
SY	11	closed	monitor environment
SY	11	closed	monitor failures

Group	Participant	Problem Type	Sub-function
SY	11	closed	monitor weather
SY	11	closed	monitor hostile fire
SY	11	closed	monitor flight
SY	11	closed	display location
SY	11	closed	control landing
SY	11	closed	guide landing
SY	11	open	control payload mode
SY	11	open	control autopilot
SY	11	open	maintain communication
SY	11	open	control emergencies
SY	11	open	control facilities
SY	11	open	transfer control
SY	11	open	manage ISR requests
SY	11	open	store health
SY	11	open	change mission status
SY	2	closed	monitor location
SY	2	closed	control attitude
SY	2	closed	detect emergencies
SY	2	closed	update route
SY	2	closed	relay locations
SY	2	closed	obtain missions
SY	2	closed	relay status
SY	2	closed	control payload mode
SY	2	closed	transfer imagery
SY	2	open	receive missions
SY	2	open	evaluate target priorities
SY	2	open	update mission plan
SY	2	open	communicate with supported units
SY	2	open	transfer control
SY	2	open	relay mission change
SY	2	open	transmit imagery
SY	2	open	receive imagery
SY	2	open	process imagery
SY	2	open	evaluate status
SY	2	open	determine status
SY	2	open	communicate with ATC
SY	2	open	monitor airspace
SY	2	open	calculate route
SY	2	open	communicate with vehicle
SY	14	closed	monitor communications
SY	14	closed	manage transponder
SY	14	closed	control surfaces
SY	14	closed	control parachute
SY	14	closed	monitor instruments

Group	Participant	Problem Type	Sub-function
SY	14	closed	monitor engine
SY	14	closed	monitor power
SY	14	closed	monitor GPS
SY	14	closed	detect emergencies
SY	14	closed	control payload mode
SY	14	closed	track target
SY	14	closed	transmit imagery
SY	14	open	prioritize targets
SY	14	open	update mission plan
SY	14	open	maintain communication
SY	14	open	monitor location
SY	14	open	monitor surfaces
SY	14	open	monitor parachute
SY	14	open	monitor instruments
SY	14	open	monitor transponder
SY	14	open	monitor engine
SY	14	open	monitor power
SY	14	open	track aircraft
SY	14	open	report location
SY	16	closed	monitor engine
SY	16	closed	monitor power
SY	16	closed	monitor autopilot
SY	16	closed	monitor instruments
SY	16	closed	monitor GPS
SY	16	closed	monitor transponder
SY	16	closed	monitor parachute
SY	16	closed	operate payload camera
SY	16	closed	control payload mode
SY	16	closed	track target
SY	16	closed	steer camera
SY	17	closed	control engine
SY	17	closed	control power
SY	17	closed	monitor instruments
SY	17	closed	control surfaces
SY	17	closed	monitor GPS
SY	17	closed	control altitude
SY	17	closed	monitor communications
SY	17	closed	monitor parachute
SY	17	closed	monitor transponder
SY	17	closed	monitor autopilot
SY	17	open	monitor mission
SY	17	open	manage imagery
SY	17	open	receive targets
SY	17	open	update mission plan

Group	Participant	Problem Type	Sub-function
SY	17	open	identify alternate landing sites
SY	17	open	assign launch/recovery site
SY	17	open	interface with HHQ
SY	17	open	assign NAI
SY	17	open	distribute ISR to RVT
SY	17	open	control navigation
SY	17	open	interface with supported units
SY	17	open	interface with ATC
SY	17	open	manage ISR requests
SY	17	open	monitor surfaces
SY	19	closed	input controls
SY	19	closed	input navigation
SY	19	closed	display location
SY	19	closed	monitor flight
SY	19	closed	control aircraft
SY	19	open	control heading
SY	19	open	control altitude
SY	19	open	sense obstacles
SY	19	open	update route
SY	19	open	monitor failures
SY	19	open	display alert
HF	1	closed	conduct pre-flight checks
HF	1	closed	obtain clearance
HF	1	closed	launch aircraft
HF	1	closed	operate payload camera
HF	1	closed	monitor engine
HF	1	closed	monitor power
HF	1	closed	monitor autopilot
HF	1	closed	monitor instruments
HF	1	closed	monitor GPS
HF	1	closed	monitor communications
HF	1	closed	monitor payload
HF	1	closed	monitor surfaces
HF	1	closed	monitor IFF
HF	1	closed	monitor parachute
HF	1	closed	monitor mechanical failures
HF	1	closed	monitor weather
HF	1	closed	monitor hostile fire
HF	1	closed	fly to area
HF	1	closed	initiate landing
HF	1	closed	initiate return
HF	1	closed	obtain clearance
HF	1	closed	conduct post flight checks
HF	1	open	transfer control

Group	Participant	Problem Type	Sub-function
HF	1	open	transfer payload
HF	1	open	monitor communications
HF	1	open	staff ASOC
HF	1	open	Staff TOC
HF	1	open	Staff ATC
HF	1	open	manage aircraft assignments
HF	1	open	manage transponder
HF	1	open	identify alternate landing sites
HF	1	open	obtain missions
HF	1	open	calculate fuel load
HF	1	open	manage call sign
HF	1	open	assign launch/recovery site
HF	1	open	assign HVT
HF	1	open	manage tail number
HF	1	open	manage ATC frequencies
HF	1	open	assign NAI
HF	1	open	assign supported unit
HF	1	open	assign task unit
HF	1	open	track aircraft
HF	1	open	manage fuel load
HF	1	open	identify alternate landing sites
HF	1	open	manage ammo
HF	1	open	update mission plan
HF	1	open	interpret imagery
HF	1	open	maintain RVT status
HF	1	open	monitor ATC
HF	1	open	notify ATC
HF	1	open	distribute ISR to RVT
HF	1	open	update mission plan
HF	8	closed	input navigation
HF	8	closed	maintain communication
HF	8	closed	communicate with ATC
HF	8	closed	communicate with GPS
HF	8	closed	conduct pre-flight checks
HF	8	closed	launch aircraft
HF	8	closed	communicate with HHQ
HF	8	closed	communicate with TOC
HF	8	closed	communicate with ASOC
HF	8	closed	monitor instruments
HF	8	closed	conduct post flight checks
HF	8	closed	initiate landing
HF	8	closed	transfer imagery
HF	8	closed	control payload mode
HF	8	closed	monitor NAI

Group	Participant	Problem Type	Sub-function
HF	8	closed	monitor parachute
HF	8	closed	monitor surfaces
HF	8	closed	monitor power
HF	8	closed	monitor instruments
HF	8	closed	monitor communications
HF	8	closed	monitor autopilot
HF	8	closed	monitor engine
HF	8	closed	monitor payload
HF	8	closed	monitor GPS
HF	8	closed	monitor transponder
HF	8	closed	update mission plan
HF	8	closed	communicate with GCS
HF	8	closed	communicate with ATC
HF	8	closed	monitor environment
HF	8	closed	monitor weather
HF	8	closed	monitor hostile fire
HF	8	open	communicate with GCS
HF	8	open	communicate with FSO
HF	8	open	update mission plan
HF	8	open	communicate with CAS
HF	8	open	communicate with GPS
HF	8	open	communicate with IFF
HF	8	open	Communicate with SOF
HF	8	open	communicate with RCTS
HF	8	open	send callsigns
HF	8	open	manage transponder
HF	8	open	manage ATC frequencies
HF	8	open	identify alternate landing sites
HF	8	open	determine status
HF	8	open	monitor RVT
HF	8	open	manage ISR requests
HF	8	open	relay locations
HF	8	open	update mission plan
HF	8	open	send CAS
HF	8	open	send CSAR
HF	8	open	Send Support Units
HF	8	open	communicate with supported units
HF	8	open	communicate with FSO
HF	8	open	monitor ISR
HF	8	open	report HVT
HF	8	open	report TAI
HF	8	open	communicate with AGCS
HF	3	closed	monitor power
HF	3	closed	monitor engine

Group	Participant	Problem Type	Sub-function
HF	3	closed	monitor autopilot
HF	3	closed	monitor instruments
HF	3	closed	monitor GPS
HF	3	closed	monitor communications
HF	3	closed	monitor payload
HF	3	closed	monitor surfaces
HF	3	closed	monitor transponder
HF	3	closed	monitor parachute
HF	3	closed	control power
HF	3	closed	control engine
HF	3	closed	control autopilot
HF	3	closed	control instruments
HF	3	closed	control GPS
HF	3	closed	maintain communications
HF	3	closed	control payload mode
HF	3	closed	control surfaces
HF	3	closed	manage transponder
HF	3	closed	control parachute
HF	3	open	identify alternate landing sites
HF	3	open	Assign Callsign
HF	3	open	Assign Transponders Codes
HF	3	open	manage fuel load
HF	3	open	Assign ATC Freq
HF	3	open	Assign TAI
HF	3	open	Assign NAI
HF	3	open	Coordinate with Tasked Unit
HF	3	open	identify landing site
HF	3	open	identify takeoff site
HF	3	open	assign HVT
HF	3	open	Determine Tail/Side Number of the Aircraft Used
HF	3	open	assign HVT
HF	3	open	Manage Supported Units
HF	3	open	assign task unit
HF	3	open	coordinate with ATC
HF	3	open	coordinate with HHQ
HF	3	open	support configuration
HF	3	open	interface with supported units
HF	3	open	transmit video
HF	3	open	maintain SA
HF	3	open	interpret imagery
HF	3	open	manage ISR requests
HF	3	open	update mission plan
HF	3	open	transfer control
HF	5	closed	proceed to waypoint

Group	Participant	Problem Type	Sub-function
HF	5	closed	monitor airspace
HF	5	closed	monitor communications
HF	5	closed	monitor fuel
HF	5	closed	monitor weapons
HF	5	closed	monitor weather
HF	5	closed	execute lost link procedure
HF	5	closed	process JTAR
HF	5	closed	process CSAR request
HF	5	closed	process SEAD request
HF	5	closed	Process ISR request
HF	5	closed	Provide imagery
HF	5	open	proceed to waypoint
HF	5	open	monitor airspace
HF	5	open	monitor communications
HF	5	open	monitor fuel
HF	5	open	monitor weapons
HF	5	open	monitor weather
HF	5	open	execute lost link procedure
HF	5	open	process CAS
HF	5	open	process CSAR request
HF	5	open	process SEAD request
HF	5	open	Process ISR request
HF	5	open	process JTAR
HF	9	closed	control heading
HF	9	closed	control altitude
HF	9	closed	control speed
HF	9	closed	receive mission plans
HF	9	closed	upload mission plans
HF	9	closed	execute mission plans
HF	9	closed	monitor mechanical failures
HF	9	closed	monitor weather
HF	9	closed	monitor fire
HF	9	closed	receive mission plans
HF	9	closed	upload mission plans
HF	9	closed	execute mission plans
HF	9	closed	control heading
HF	9	closed	control altitude
HF	9	closed	control speed
HF	9	closed	track motion
HF	9	closed	steer camera
HF	9	closed	monitor mechanical failures
HF	9	closed	monitor weather
HF	9	closed	monitor fire
HF	9	closed	control engine

Group	Participant	Problem Type	Sub-function
HF	9	closed	control power
HF	9	closed	control autopilot
HF	9	closed	control instruments
HF	9	closed	control GPS
HF	9	closed	control communications
HF	9	closed	control surfaces
HF	9	closed	control transponder
HF	9	closed	control parachute
HF	9	closed	control heading
HF	9	closed	control altitude
HF	9	closed	control speed
HF	9	closed	receive mission plans
HF	9	closed	upload mission plans
HF	9	closed	execute mission plans
HF	9	closed	monitor mechanical failures
HF	9	closed	monitor weather
HF	9	closed	monitor fire
HF	9	open	manage callsigns
HF	9	open	manage codes
HF	9	open	determine tasked unit
HF	9	open	identify landing site
HF	9	open	identify takeoff site
HF	9	open	identify alternate landing sites
HF	9	open	plan missions
HF	9	open	enable handoffs
HF	9	open	transfer control
HF	9	open	process data
HF	9	open	maintain communications
HF	9	open	manage frequencies
HF	9	open	manage fuel load
HF	9	open	manage missions
HF	9	open	track aircraft
HF	9	open	manage fuel load
HF	9	open	manage ammo
HF	9	open	manage facility
HF	9	open	interpret imagery
HF	9	open	maintain SA
HF	9	open	monitor fuel
HF	9	open	Provide communications
HF	9	open	provide CAS
HF	9	open	provide route clearance
HF	9	open	support infantry
HF	9	open	support SOF
HF	9	open	support CSAR

Group	Participant	Problem Type	Sub-function
HF	9	open	Support Fires
HF	9	open	monitor mission
HF	9	open	track tail number
HF	9	open	watch HVT
HF	9	open	Determine Tail/Side Number of the Aircraft Used
HF	9	open	control NAI
HF	9	open	monitor units
HF	15	closed	monitor engine
HF	15	closed	monitor GPS
HF	15	closed	monitor power
HF	15	closed	monitor instruments
HF	15	closed	monitor heading
HF	15	closed	control surfaces
HF	15	closed	monitor transponder
HF	15	closed	operate payload camera
HF	15	closed	control payload mode
HF	15	closed	maintain communications
HF	15	closed	monitor engine
HF	15	closed	monitor GPS
HF	15	closed	control parachute
HF	15	open	identify callsigns
HF	15	open	identify transponder codes
HF	15	open	allocate assigned missions
HF	15	open	identify HVT
HF	15	open	identify TAI
HF	15	open	identify NAI
HF	15	open	identify supported units
HF	15	open	identify tasked unit
HF	15	open	identify takeoff site
HF	15	open	identify landing site
HF	15	open	identify alternate landing sites
HF	15	open	identify fuel load locations
HF	15	open	assign frequencies
HF	15	open	Determine Tail/Side Number of the Aircraft Used
HF	15	open	track aircraft
HF	15	open	monitor fuel
HF	15	open	monitor ammo
HF	15	open	manage facility
HF	15	open	monitor lost personnel
HF	15	open	deconflict missions
HF	15	open	monitor payload
HF	15	open	interface with ATC
HF	15	open	transfer control
HF	15	open	manage requests

Group	Participant	Problem Type	Sub-function
HF	15	open	update mission plan
HF	20	closed	enable communications
HF	20	closed	monitor mission
HF	20	closed	update mission plan
HF	20	closed	get destination
HF	20	closed	determine position
HF	20	closed	update location
HF	20	closed	calculate nav methods
HF	20	closed	control surfaces
HF	20	closed	control power
HF	20	closed	read mission objective
HF	20	closed	read mission constraints
HF	20	closed	monitor weather
HF	20	closed	identify target
HF	20	closed	track target
HF	20	closed	control payload mode
HF	20	open	switch control mode
HF	20	open	receive commands from operator
HF	20	open	interpret imagery
HF	20	open	transfer imagery
HF	20	open	collect state data
HF	20	open	read state data
HF	20	open	monitor aircraft
HF	20	open	control surfaces
HF	20	open	activate communications
HF	20	open	monitor communications
HF	20	open	interpret communications
HF	20	open	control surfaces
HF	20	open	determine position
HF	20	open	identify ATC
HF	20	open	read ATC commands
HF	20	open	request ATC commands
HF	20	open	read requests
HF	20	open	prioritize requests
HF	20	open	deconflict missions
HF	20	open	disseminate flight plans
HF	20	open	read mission updates
HF	20	open	execute mission plans