Towards a Framework for Architecting Heterogeneous Teams of Humans and Robots for Space Exploration

by

Julie Ann Arnold

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author .....
Department of Aeronautics and Astronautics
May 26, 2006

Certified by .....
Jeffrey A. Hoffman
Professor of the Practice of Aerospace Engineering
Thesis Supervisor

Certified by .....
Joseph H. Saleh
Executive Director of the Ford-MIT Alliance
Thesis Supervisor

Accepted by .....
Jaime Peraire
Chairman, Department Committee on Graduate Students

Towards a Framework for Architecting Heterogeneous Teams of Humans and Robots for Space Exploration

by

Julie Ann Arnold

Submitted to the Department of Aeronautics and Astronautics
on May 26, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Human-robotic systems will play a critical role in space exploration, should NASA embark on missions to the Moon and Mars. A unified framework to optimally leverage the capabilities of humans and robots in space exploration will be an invaluable tool for mission planning. Although there is a growing body of literature on human robotic interactions (HRI), there is not yet a framework that lends itself both to a formal representation of heterogeneous teams of humans and robots, and to an evaluation of such teams across a series of common, task-based metrics. My objective in this thesis is to lay the foundations of a unified framework for architecting human-robotic systems for optimal task performance given a set of metrics. First, I review literature from different fields including HRI and human-computer interaction, and synthesize multiple considerations for architecting heterogeneous teams of humans and robots. I then present methods to systematically and formally capture the characteristics that describe a human-robotic system to provide a basis for evaluating human-robotic systems against a common set of metrics. I propose an analytical formulation of common metrics to guide the design and evaluate the performance of human-robot systems, and I then apply the analytical formulation to a case study of a multi-agent human-robot system developed at NASA. Finally, I discuss directions for further research aimed at developing this framework.

Thesis Supervisor: Jeffrey A. Hoffman
Title: Professor of the Practice of Aerospace Engineering

Thesis Supervisor: Joseph H. Saleh
Title: Executive Director of the Ford-MIT Alliance

Acknowledgments

I am very fortunate to have had two incredible advisors in the course of doing this work, both of whom have taught me how to be a researcher and inspired me to reach for my full potential.

Professor Jeffrey Hoffman met with me nearly every week to help steward my research and talk about academics. Through these conversations he encouraged me to explore research topics that interested me most, and helped give focus and direction to my ideas. I can not imagine a better advisor to guide me in the transition from the undergraduate to the graduate world and beyond. I am unendingly grateful for the opportunities Jeff has passed my way (the most memorable being an expedition to the Arctic!) In addition to being a great academic advisor, he is a wonderful mentor and friend to me.

Joseph Saleh is an endless source of ideas and inspiration. When he gave a talk to the MVL in Fall 2004, I could not wait to work with him. Since then, he has guided me through the daunting task of researching and preparing my first publications. I most appreciated his ability to point me in the right direction each time I began feeling lost. He has offered a helping hand in nearly every aspect of my graduate education - he even offered me his controls notes when I TAed 16.06! I am very grateful for his major involvement in this work, his encouragement, and his advice in all areas.

I would also like to thank Professor Dava Newman, Professor John Deyst, Professor Karen Wilcox, Professor Olivier de Weck, and Professor Brian Williams for their interest in my professional development and their invaluable advice.

I greatly appreciated the opportunity to collaborate with Jean Scholtz from the National Institute for Standards and Technology (NIST) and Terry Fong from NASA Ames Research Center.

My lab mates at the MVL have made the past two years fly by. I would especially like to thank my office mates Nicole and Chris for often lending their ears and advice.

Lastly, I would like to thank my Mom and Dad for encouraging every dream I

had as a kid, my sister Diana for her constant enthusiasm, and my fiancee Neel for his love and support (and gourmet cooking).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As NASA prepares to embark on exploration missions to the Moon and Mars, human-robotic systems will play a critical role in fulfilling mission objectives. Humans and robots will be required to work in teams to explore and conduct science on planetary surfaces, and erect, maintain, and repair space-based infrastructure. NASA has made a great deal of progress towards developing and demonstrating robotic and automated technologies over the past decade, and current research activity is focused on testing, demonstrating, and validating specific robotic and automated technologies that will enable effective human-robotic systems [19, 44, 47, 51, 80, 83, 13]. However, these technologies are evaluated using a wide range of metrics that are highly application specific. There is no existing framework of common metrics to compare the advantages and disadvantages of different human-robotic systems performing a space exploration task or set of tasks. Formulating such a framework is a necessary step towards optimally leveraging the capabilities of humans and robots in space exploration.

In this thesis, I develop the foundation for such a framework through a review and synthesis of considerations for architecting human-robot teams (Chapter 2). To structure the discussion I propose an outline of a framework for designing human-robotic systems consisting of four building blocks: 1) specifying tasks, 2) generating a human-robotic team, 3) allocating functions to agents in the system, and 4) evaluating the system against common task-based metrics. I then use this structure to discuss considerations for designing optimal human-robotic systems to perform a task or set

of tasks, given a set of metrics.

In Chapter 3, I present further work towards developing formal methods to represent human-robotic system architectures. My objective is to provide a basis for a standard means of evaluating human-robotic systems against a common set of metrics. Evaluating different human-robotic systems based on a non-standard description of system architecture makes it difficult to compare the evaluation results for different systems. Rather, a designer must systematically and formally capture the characteristics that describe each human-robotic system to ensure that the evaluations of different human-robotic systems are comparable. In this chapter, I discuss the important elements that must be captured in the formal representation of a human-robotic system. I then illustrate the formal representation through a case-study of the terrestrial human-robotic Nursebot System, which provides assistance to nursing home residents and caregivers.

In Chapter 4, I propose an analytical formulation of common metrics to guide the design and evaluate the performance of human-robot systems. The increased relevance of human-robot systems raises the issue of how to optimally (and reliably) design these systems to best leverage the varied capabilities of humans and robots. The question of optimality in turn raises the question of what metrics to use in order to guide the design, and evaluate the performance, of human-robot systems. In this chapter, I present objectives for maximizing the effectiveness of a human-robot system which capture the coupled relationships among productivity, reliability, and risk to humans. Reliability parameters are proposed to characterize unplanned interventions between a human and robot, and the effect of unplanned interventions on the effectiveness of human-robot systems is then investigated analytically using traditional reliability analysis.

I then present a real-world example of carrying out the theoretical analysis of the effectiveness of a human-robot system (Chapter 5). I apply the analytical formulation from Chapter 4 to a case-study of the Peer-to-Peer System, a multi-agent human-robot system developed at NASA. I describe the Peer-to-Peer System, including agents and their roles, and a formal description of the tasks performed during experiment trials.

I then use data collected during the experiment trials to compare the effectiveness of the system to a human-only team performing the same tasks.

Finally, in Chapter 6, I summarize the contributions of this thesis towards a unified framework for analytically comparing the advantages and disadvantages of human-robot systems. I also discuss directions for future research, and potential applications of such a framework in space exploration and other fields.

Chapter 2

Review and synthesis of considerations for design and evaluation of human-robot multi-agent systems

New robotic and automated technologies provide many options for architecting teams of humans and robots for space exploration. However, the nature of space exploration requires that human-robotic systems perform tasks in changing environments that are difficult to characterize. It is an open question how to best integrate these technologies to optimally leverage the capabilities of humans and robots for specific space exploration tasks. In this chapter, I undertake the analysis of considerations for architecting human-robotic systems from a designer's perspective to facilitate the future transition from analysis to design of human-robotic systems.

[1]This chapter is based on work under consideration for publication with IEEE Transactions on Systems, Man, and Cybernetics.

2.1 Analyzing human-robotic systems from a designers' perspective

In Figure 2-1 I propose a framework for designing human-robotic systems to perform a task or set of tasks. I use this framework to guide the discussion of considerations for architecting heterogeneous teams of humans and robots for optimal task performance, given a series of tradeoffs among various metrics. The building blocks of this framework include 1) specifying tasks, 2) generating a human-robotic team, 3) allocating functions to agents in the system, and 4) evaluating the system against common task-based metrics. A designer must first identify the tasks to be performed by the human-robotic system and define the environment in which these tasks will be performed. Tasks must be defined independent of the human-robotic system performing them in order to provide for a framework in which human-robotic systems are comparable across a series of common, task-based metrics. Task specification must also capture the uncertainty of the environment in which the task or tasks are performed; this lays the foundation for a framework capable of addressing the value of flexibility and robustness in human-robotic systems. The specification of tasks and environment can be used to formulate a set of common metrics to compare different teams of humans and robots.

Next the designer must formally describe the human-robotic system architecture. This includes defining the number and type of agents, capabilities and constraints of each agent, and the ways in which agents interact (including hierarchical relationships). The designer then searches the space of feasible options for allocating tasks to members of the team architecture. This search yields an optimal task allocation for a given set of common, task-based metrics. The designer then formulates a new team architecture and iterates through the design process until all architectures of interest have been evaluated for comparison. Finally, the designer compares each of the team architectures (with optimal task allocation), and chooses the team architecture which best meets the designer's objectives.

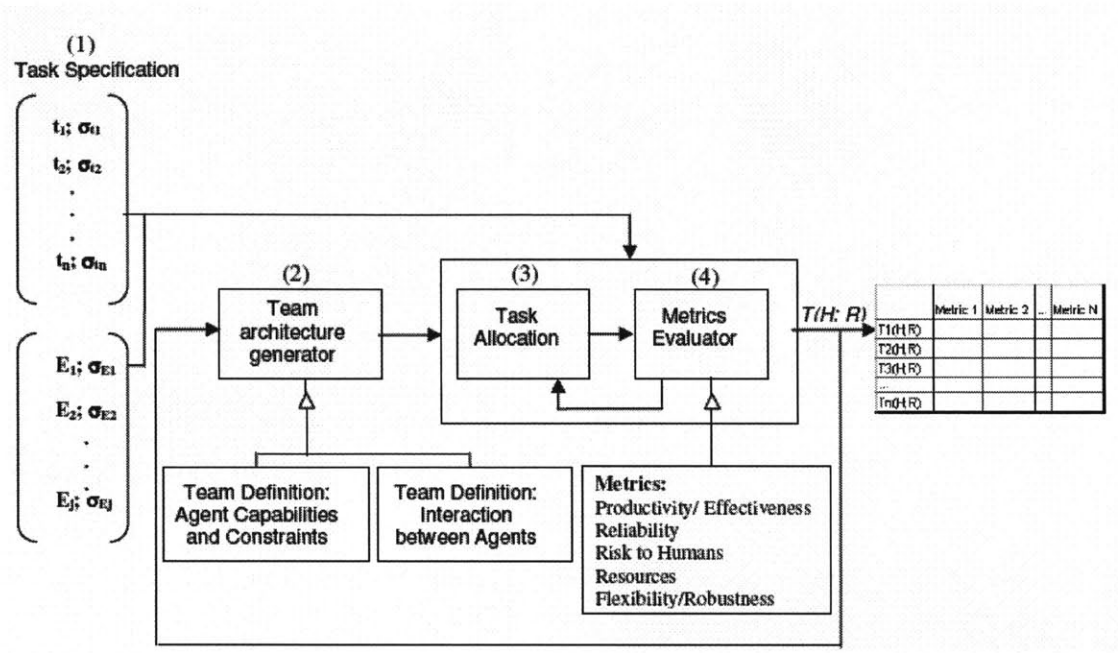

Figure 2-1: Flowchart of framework for designing optimal human-robotic systems

While there is a great deal of literature addressing each of the building boxes in the proposed framework, these works have not yet been integrated into a unified framework for designing optimal human-robotic systems to perform a task or set of tasks, given a set of metrics. In this chapter, I investigate considerations relevant to each numbered building block of the framework proposed in Figure 2-1. I discuss issues related to (1) task specification (Section 2.2). I also discuss considerations related to (2) formally describing a team architecture. The modes in which humans and robots can work together to execute a space exploration task or set of tasks depend on considerations such as inherent capabilities and constraints of each agent and the different modes in which agents interact (Section 2.3). I explore desirable characteristics of (3) task allocation methods for a team architecture (Section 2.4). Finally, I propose (4) common metrics to compare different teams of humans and robots (Section 2.5).

2.2 Task specification

The first step in designing a human-robotic system (depicted as the first building block in Figure 2-1 is to describe the tasks to be performed and the environment in which they will be performed. One must adequately capture the characteristics of the task and environment that are anticipated to potentially influence system performance. This provides a basis for evaluating the performance of the human-robotic system [12]. In this section, first I outline the characteristics a description of task and environment should capture. I then draw on the literature to discuss the different methods of task specification. From this literature review, I synthesize and propose a method to specify tasks and environment. Finally, I present an illustrative example of the synthesized task specification method.

2.2.1 Characteristics of task specification

Specification of tasks must be independent of the human-robotic system performing it in order to provide for a framework in which human-robotic systems are comparable across a series of common, task-based metrics. Common task-based metrics are necessary to compare the performance of different human-robotic systems performing the same tasks. Task specification must also characterize the uncertainty of the environment in which the task or tasks are performed; this will lay the foundation for a framework capable of addressing the value of flexibility and robustness in human-robotic systems.

2.2.2 Specifying tasks: literature review

Often, descriptions of tasks for robotic systems or human-robotic systems are system-specific. This leads to system-specific metrics for evaluating the human-robotic system and precludes the ability to compare the advantages and disadvantages of different human-robotic systems performing the same task(s). Potkonyak et al. [60] remind us that formulating the task specification with a description of the final outcome of the task opens up the possibility for multiple solutions to achieve the desired

outcome. Specifying a task so that multiple human-robotic systems can fulfill the final outcome is a necessary step towards formulating common, task-based metrics which can be used to compare the advantages and disadvantages of different human-robotic systems. The interested reader is referred to literature on system architecture theory for further discussion of system-independent task and requirements specification [24].

Once we have specified the final outcome of the task(s) for which we are designing a human-robotic system, we require a method of further specifying the subtasks that allow for the task(s) to be successfully completed. These subtasks must be system-independent and describe the task to be performed rather how a particular system performs the task. This taxonomy of system-independent task specification and system-dependent task specification is addressed in the field of assembly and task planning [38] and is also relevant for human-robotic systems operating in the space environment. System-independent task specification is captured in the first building block in Figure 2-1 and is the subject of discussion in this section. System-dependent task specification is a property of a particular human-robotic system and is discussed further in Chapter 3.

The literature provides us with a number of different methods for system-independent task specification. In a discussion of task allocation for human-robot interaction in manufacturing, Ghosh et al. [37] describe a method for defining tasks by dividing tasks into subtasks and listing general and specific requirements for completing each subtask. An example of Ghosh et al. task specification is presented in Table 2.1.

Rodriguez et al. [62] propose a similar method in a paper discussing human-robot system performance for space exploration. Tasks are decomposed into functional primitives and associated performance metrics. An example of Rodriguez et al. task specification is shown in Table 2.2.

However, Acquisti et al. [1], suggest that these types of functional analyses are limited in that they do not consider "informal logistics" such as environmental conditions and problem resolution. (See [42, 17] for discussion of the environmental conditions relevant in exploring planetary surfaces.) The nature of space exploration requires that human-robotic systems perform tasks in changing environments which are diffi-

Task: Assembly Operation		
Subtask	**General Requirements**	**Specific Requirements**
Reach	.....	.....
Select	.....	.....
Grasp	.....	.....
Move	Degree of Freedom	Weight of Object
		Distance
		Speed of Movement
		Acceleration
Position	.....	.....
Assembly	.....	.....

Table 2.1: Ghosh et al. task specification [37]

Task: Collect Geologic Sample	
Functional Primitive	**Performance Metrics**
Traverse	Distance Traveled
	Terrain Degree of Difficulty
Find Rocks	.....
Carry Rocks	.....

Table 2.2: Rodriguez et al. task specification [62]

cult to fully characterize, and the capability for flexibility and robustness become key drivers in evaluating a human-robotic system operating in a changing environment. As the environment changes, same actions will have different effects. Therefore, the actions taken by agents in a human-robotic system are dependent on the context in which the actions are being performed. This phenomenon is referred to as context-conditioned variability, unanticipated variability, or situated action, and is recognized in a variety of fields including motor control, cognitive engineering, and cognitive science [78]. The interested reader is referred to [49, 9, 75] for further discussion of environmental context and situated action.

Consequently, we require a task specification that is capable of capturing the information about the environment and state of the human-robotic system. In addressing this requirement, we review work on artificial intelligence, which addresses the problem of capturing temporal relationships among actions and events [3, 71, 18, 53]. For example, Nicolescu et al. [53] propose a task specification which captures this infor-

mation in the form of a behavior network. A behavior network is a link-node representation of context-conditioned, complex sequences of behaviors. Although Nicolescu et al. implement these behavior networks to help robots learn representations of high-level tasks, they capture the characteristics we require for task specification; they use precondition/postcondition dependencies to allow for task specification that is dependent on environmental and system states [53]. Table 2.3 shows an example of implementing Nicolescu's behavior network framework to specify tasks.

Task: Collect Geologic Sample			
Behavior	**Permanent Preconditions**	**Enabling Preconditions**	**Ordering Constraints**
Traverse	Traverse mechanisms functional	....	....
	No obstacles larger than X[units]		
Find Rocks	Find Rocks mechanisms functional	Area traversed has rocks	....
Carry Rocks	Carry Rocks mechanisms functional	Rocks to carry must be less than Y [units]	Must Find Rocks before Carry Rocks

Table 2.3: Example of behavior network task specification adapted from [53]

Each behavior (or subtask) is described by permanent and enabling preconditions, and ordering constraints. Permanent preconditions must be met during the entire execution of the behavior or else the behavior is not possible. When a rover is traversing a boulder-strewn field, one permanent precondition is that the boulders over the course of the entire traverse must be small enough for the rover to negotiate. If the rover encounters a boulder too large to negotiate, it is not possible to continue traversing. Enabling preconditions must be met immediately before the activation of a behavior but do not need to be met during the entire execution of the behavior. If a rover is tasked with finding rocks, the enabling precondition is that there are rocks in the area. This is an enabling precondition because the rover can continue to carry out the action of finding rocks for some time after all rocks in the area have been surveyed. Finally, ordering constraints specify the sequences of behaviors. For example, it is necessary to find rocks before carrying rocks.

Once we have specified tasks dependent on environment and system states, we

must represent uncertainties in the tasks to be performed and the environmental and system states. By capturing these uncertainties in task specification, we lay the foundation for evaluating the flexibility/robustness of a system, defined as the ability of a human-robotic system to accommodate uncertainties in task and environment. (See Section 2.5 on common metrics for further discussion of flexibility/robustness.) The Acquisti et al. [1] Brahms Activity Model of work practices aboard the International Space Station represents uncertainties in tasks to be performed by categorizing tasks and subtasks according to a two by two matrix. On one axis, the task or subtask is scored according to the degree to which it is scheduled; on the other axis, the task or subtask is scored according the uniqueness or repeatability of the activity. This sort of categorization assigns a specific uncertainty to each task or subtask. I propose implementing a similar description of uncertainty to each task and subtask in the task specification. This description can be formulated as a probability function, or a quantitative or qualitative categorical ranking. The ability of human-robotic systems to accommodate uncertainty in the tasks and subtasks to be performed provides a measure of the system flexibility. Flexibility as a metric for system performance is discussed further in Section 2.5.

The Brahms Activity Model also links tasks to environment and system states and investigates the effect that environmental and system states, such as background noise, have on performance. To evaluate and compare the performance of different human-robotic systems for space exploration tasks with the framework proposed in Figure 2-1, I assert it is necessary to describe the uncertainty in the environmental and system states. For example, it may be important to evaluate how different human-robotic systems perform with the mean-level of background noise as well as the high and low bounds for level of background noise. This description of uncertainty in environmental and system states can be formulated as a probability function, or a quantitative or qualitative categorical ranking. The ability of human-robotic systems to accommodate uncertainty in the tasks and subtasks to be performed provides a measure of the system robustness. Robustness as a metric for system performance is discussed further in Section 2.5.

By directly linking tasks and subtasks with environment conditions, subtask ordering constraints, and associated uncertainties, the goal is to synthesize a task specification method that will ultimately provide the capability to evaluate the performance of human-robotic system operating in an uncertain and changing environment.

2.2.3 Synthesized method for task specification

From this discussion I now synthesize and outline a method of defining tasks and environment. The method, consisting of the four steps justified in the previous section, are outlined below and illustrated in Figure 2-2:

1. Identify final outcome of task or set of tasks to be performed
2. Identify sub-tasks (or functional primitives) which are independent of the choice of human-robotic system
3. For each sub-task or functional primitive, identify precondition and post-condition dependencies with environment and system states
4. Describe uncertainty associated with subtasks and environmental states

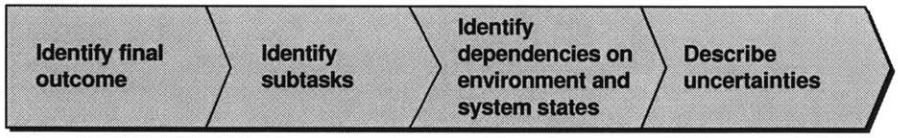

Figure 2-2: Outline of synthesized method for task specification

2.2.4 Illustrative example of synthesized method for task specification

In Tables 2-4 and 2-5 I present an illustrative example of applying this method to the Intelsat-VI Repair task performed by STS-49. The Intelsat-VI Repair is a particularly interesting example of the effect uncertainty in environment and system states can

have on even simple space tasks. During launch of the Intelsat VI in 1990, the second stage failed to separate from the satellite, resulting in an orbit which was too low and unstable to be of use. NASA eventually decided to salvage the satellite with a Space Shuttle repair mission to capture Intelsat-VI and attach a kick motor to the satellite to boost it up to its intended orbit. However, attempts by an EVA astronaut to capture the satellite failed. The capture bar failed to engage and the satellite began to wobble in a conical spin. Ultimately three EVA astronauts were required to capture and stabilize the satellite by hand [4].

I apply the synthesized method for task specification on the Intelsat-VI repair task as an illustrative example to motivate the importance of capturing uncertainty in environment and system states at the level of task specification.

In this example, *Subtasks Functions* are the actions required to carry out the task of providing Intelsat-VI the capability to boost from LEO to GEO. Each subtask function is defined to be independent of the choice of human-robotic system performing the task. Each subtask function also has an associated *Subtask Uncertainty*, describing the likelihood that each subtask function will be performed. In this particular example, there is no uncertainty with the subtask functions to be performed.

Each subtask function is performed in the context of environmental states. *Permanent Precondition States* must be met during the entire execution of the subtask or else the subtask is not possible. *Enabling Precondition States* must be met immediately before the activation of a subtask but do not need to be met during the entire execution of the subtask. Also, each permanent and enabling precondition environmental state has an associated *Environmental Uncertainty* describing the likelihood that state will be fully realized, partially realized, or not realized. This description of likelihood can be for example: a probability mass function (as shown in the illustrative example), probability distribution, or qualitative categorical ranking (for example: high, medium, low).

In this section I have discussed considerations relating to task specification, the first building block in the framework proposed for architecting human-robotic systems for space exploration tasks (Figure 2-1). I outlined the characteristics a description

Table 2.4: Illustrative example of synthesized task specification method

Final Outcome of Portfolio of Tasks: Provide Intelsat capability to boost from LEO to GEO								
SUBTASKS		**ENVIRONMENT**				**ORDERING CONSTRAINTS**		
Function	**Uncertainty**	**Permanent Preconditions**		**Enabling Preconditions**		**Pre**	**Post**	**Concurrent**
		State	**Uncertainty**	**State**	**Uncertainty**			
Position For Capture	None	Intelsat position known	Z% likelihood within Y [units of distance]	Shuttle is in LEO within X [units distance] of Intelsat	None	None- first task to be performed	Engage Capture with Intelsat	Document Task
			B% likelihood within V [units of distance]					
		Intelsat orbit, trajectory known	F% likelihood that trajectory is known within L [units of distance]					
			G% likelihood that trajectory is not known within L [units of distance]					
Engage Capture with Intelsat	None	Intelsat position known	Q% likelihood within E [units of distance]	Capture bar is positioned "optimally"	J% within I [units from optimal]	Position Capture Bar	Transport Intelsat to servicing structure	Document Task
			A% likelihood within L [units of distance]					
		Intelsat orbit, trajectory known	O% likelihood that trajectory is known within B error					
			M% likelihood that trajectory is not known within B error					
		Intelsat dynamic behavior known	S% likelihood that behavior is known within N error		K% within AA [units from optimal]			
			V% likelihood that behavior is not known within N error					
		Capture mechanism on Capture Bar functions as expected	T% likelihood it functions as expected					
			R% likelihood it does not function as expected					
Transport Intelsat to servicing structure	None	Intelsat dynamic behavior known	BB% likelihood that behavior is known F error	Intelsat position before transport is known	CC% likelihood that Intelsat position is known within DD [units of distance]	Engage Capture with Intelsat	Replace Intelsat rocket motor	Document Task
			P% likelihood that behavior is not known F error					
		Position of servicing structure known	C% likelihood within EE [units of distance]					

			FF% likelihood within U [units of distance]					
		Capture mechanism of servicing structure functions as expected	GG% likelihood it functions as expected		HH% likelihood that Intelsat position is known within II [units of distance]			
			JJ% likelihood it does not function as expected					
		Intelsat is docked with transfer mechanism	KK% likelihood it is and remains docked during subtask					
			LL% likelihood it is not docked, or undocks, during subtask					
Replace Intelsat rocket motor	None	Intelsat remains secured in servicing structure	MM% likelihood it remains secured	Not Applicable		Transport Intelsat to servicing structure	Release Intelsat from servicing structure	Document Task
			NN% likelihood it becomes unsecured					
		Attachment mechanism functions as expected	OO% likelihood that it functions as expected					
			PP% likelihood that it does not function as expected					
Release Intelsat from servicing structure	None	Release mechanism of servicing structure functions as expected	H% likelihood it functions as expected	Position of Intelsat in servicing structure known	QQ% likelihood that Intelsat position is known within RR [units of distance]	Replace Intelsat rocket motor	None- end of task	Document Task
			SS% likelihood it does not function as expected					
		Intelsat dynamic behavior known	N% likelihood that behavior is known within TT error		UU% likelihood that Intelsat position is known within VV [units of distance]			
			WW% likelihood that behavior is not known within XX error					
		Intelsat trajectory known	YY% likelihood that trajectory is known within ZZ error					
			AB% likelihood that trajectory is not known AC error					
Document Task	None	All States and States Uncertainties listed above		None	None	None	None	Document Task
		Documenting mechanisms functional	D% likelihood mechanisms functional					
			W% likelihood mechanisms not functional					

Table 2.5: Illustrative example of synthesized task specification method, cont.

of task and environment should capture and have drawn on the literature to discuss the different methods of task specification. The interested reader is referred to [37, 62, 78, 53, 70] for more detailed discussions of task specification techniques. From this literature review, I have synthesized and proposed a method to specify tasks and environment and presented an illustrative example of our task specification method. Next, I will discuss considerations related to the second building block in the proposed framework: formally describing human-robotic systems.

2.3 Option space available for architecting human-robotic systems

Once tasks have been specified, the designer generates multiple human-robotic system architectures capable of performing these tasks. This is the second building block in the framework proposed in Figure 2-1. In this section, I describe considerations for specifying what constitutes a team architecture, and how to generate multiple team architectures. Before generating architectures, it is important to consider how to formally represent human-robotic system architectures. First the designer must bound the option space by formally specifying the range of capabilities and constraints for each type of agent. Then the designer must formally specify the modes in which agents interact as a team. The designer iterates through this process to generate multiple architectures for human-robotic systems. In this section, I draw from the literature to discuss considerations related to specifying capabilities and constraints for each agent and the modes in which agents interact. I then describe how these considerations can be used to generate multiple team architectures.

2.3.1 Specifying capabilities and constraints of agents

In specifying the range of capabilities and constraints of each type of agent in a human-robotic system, I address the following questions: What are the types of tasks or functions the agent is capable of performing? What are the limitations or con-

straints of the agent in performing these tasks or functions? What is each agent's level of autonomy? From a designer's perspective, these considerations specify the current or anticipated state of technology available in designing an agent for a human-robotic system. In this section I discuss considerations related to answering each of these questions.

Embodiment of an agent

The first two of these questions relate to the embodiment of the agent, or how the physical body of the agent interacts with the environment and system states through the constraints of the body [46]. Embodiment is a concept that is relevant to cognitive science research in artificial intelligence, artificial life, and robotics [52]. Quick et al. [61] provides a formal definition for embodiment:

> "A system X is embodied in an environment E if pertubatory channels exist between the two. That is, X is embodied in E if for every time t at which both X and E exist, some subset of E's possible states have the capacity to perturb X's state, and some subset of X's possible states have the capacity to perturb E's state."

Moreover, embodiment is not necessarily a binary attribute. Although all agents are embodied, some may exhibit a higher degree of embodiment than others [52, 61, 21]. "For instance, the greater the perturbatory 'bandwidth' connecting agent and environment, the higher the degree of embodiment [52]."

I delineate two categories of embodiment: physical and cognitive. The physical body of the agent allows both physical and cognitive interaction with the environment and system states. The ways in which the body of the agent physically interacts with the environment is a function of its inherent physical abilities. For example, the Space Shuttle robotic arm known as the Remote Manipulator System (RMS) is capable of physically interacting with the environment and system states by moving in three-dimensional space and grappling objects. The ways in which the body of the agent cognitively interacts with the environment and system states is a function

of its inherent cognitive abilities. For example, the RMS is able to cognitively interact with its environment and system states by continuously monitoring its own state-information (or "health") for changes that may indicate, for example, collision or contact with an object in the environment [73]. While the RMS is continuously monitoring its own state-information, some subset of the environment's states (such as the location of other objects) has the capacity to perturb the RMS's states (such as position and orientation), and vice versa. Perturbations in either the RMS's states or the environment states result in a perturbation in the sensor information relayed to the RMS and represent a cognitive interaction with the environment.

Autonomy of an agent

Specifying the level of autonomy of an agent is a more complicated matter. There is a great deal of literature addressing the definition of autonomy [40, 39, 35, 7, 77, 10]. However, to date no universally accepted definition of autonomy exists. In this section I briefly present a sampling of definitions for autonomy and motivate the choice of the definition of autonomy we use in this work. I then elaborate on the considerations related to my choice of definition for autonomy.

Franklin et al. [35] propose the following definition for autonomy based on a survey of various agents:

> "An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

While this definition captures the importance of environment in a formal definition of autonomy, it describes autonomy as an absolute quantity unrelated to the degree to which an agent is free from intervention by other agents. Barber et al. [7] addressed this shortcoming in the following proposed definition for autonomy:

> "An agent's degree of autonomy, with respect to some goal that it actively uses its capabilities to pursue, is the degree to which the decision-

making process, used to determine how that goal should be pursued, is free from intervention by any other agent."

However, the ultimate aim of Barber et al. [7] is to quantify autonomy, and as a result their definition overlooks the underlying *cause* of the degree to which an agent is free from intervention by any other agent: the agent's ability to sense and act in a changing environment.

For this work I adopt and build on the following basic definition of **autonomy: the ability of an agent to accommodate variations in the environment (in pursuit of its goals) [77].** When we discuss autonomy in the context of the framework proposed in Figure 2-1, we are primarily interested in understanding the capabilities and constraints of agents in a human-robotic system to sense and act in a changing environment. The nature of space exploration requires that human-robotic systems perform tasks in changing environments which are difficult to fully characterize and the capability for flexibility and robustness are key drivers in evaluating a human-robotic system. Specifying the autonomy of agents as the ability to accommodate variations in the environment provides the ability to describe the capabilities and constraints of agents as they relate to the task and environment uncertainties described in task specification (Section 2.2), and lays a foundation for ultimately evaluating the performance of human-robotic systems operating in an uncertain and changing environment.

I propose that there are two components to this definition of autonomy: agent and team. Each agent in a human-robotic system has its own inherent level of ability to accommodate variations in the environment in pursuit of its goals. However, team interaction may augment or diminish this inherent level of ability. (This proposed taxonomy is inspired by the quantitative treatment of relative autonomy for multi-agent interaction presented in [10].) For example, imagine there are no variations in environment in our task specification and we know exactly where the satellite is in relation to Space Shuttle Remote Manipulator Arm. In this case the Remote Manipulator Arm is able to successfully navigate to and grapple the satellite without any interaction with other agents. However, if the satellite is actually two meters to

the left of the expected position, the Remote Manipulator Arm would not be able to navigate to and grapple the satellite without help from other agents. Thus the Remote Manipulator Arm has a low level of agent autonomy. However, with a human in the loop providing information about the actual position of the satellite, the Remote Manipulator Arm can accommodate a great deal of variation in the position of the satellite (i.e. variation in the environment). Thus a certain level of team interaction between a human and the Remote Manipulator Arm increases the Arm's autonomy (or ability to accommodate variation in the environment). The team component of autonomy is primarily a function of interaction among agents, and considerations related to interaction among agents are discussed in Section 2.3.2.

2.3.2 Specifying modes of interaction among agents

Pertaining to the second building block in the framework proposed in Figure 2-1, I have discussed considerations related to specifying the inherent capabilities and constraints of agents in a human-robotic system. I now discuss considerations related to how these agents may interact. There are at least two characteristics necessary to describe the interaction between agents in a human-robotic system: the level of cognitive interaction between agents, and level of physical interaction between agents [7, 31].

Cognitive Interaction

While the Human-Robot Interaction (HRI) community has made progress toward defining common metrics for evaluating human-robot interactions [31, 20], I did not find a consensus on or an explicitly categorized list of the different modes of cognitive interaction. As a result, I present characteristics of cognitive interaction synthesized from sources addressing interaction with automation, metrics for human-robotic interaction, and studies on human-robotic interaction. These characteristics are decomposed to form a set of independent cognitive interaction modes which can be used to describe the cognitive interaction option space available for architecting heterogeneous teams of humans and robots.

Characteristics of cognitive interaction: literature review

As a first step in developing modes of cognitive interaction, I discuss what types of interaction are applicable from human interaction with automation, addressed in the field of human-machine interaction (HMI). Parasuraman et al. [57] propose four classes of functions for automation: information acquisition, information analysis, decision and action selection, and action implementation. In my framework, action implementation is considered a property of task allocation rather than cognitive interaction. However, the other three functions are characteristics of cognitive interaction and I propose that they can be generalized to modes of cognitive interaction between any two agents:

- **Information Exchange (IE):** characterizes the flow of information between two agents in terms of agent requests and transfer of input
- **Information Assessment (IA):** characterizes what state and environment information an agent is able to gather and assess as nominal (according to plan) vs. off-nominal (not according to plan) about other agents.
- **Decision and Action Selection (DS):** characterizes how two agents work (or do not work) together to make execution decisions. ([34, 8] propose taxonomies for this mode of interaction)

While the four functions for automation proposed by Parasuraman et al. may be sufficient in describing human interaction with automation, human-robot interaction is fundamentally different from HMI because "it concerns systems which have complex, dynamic control systems, exhibit autonomy and cognition, and which operate in changing real-world environments. [69]" As a result, one would expect that the three modes of cognitive interaction derived from HMI are not sufficient to describe all modes of agent-agent cognitive interaction for our framework.

I review HRI sources with the purpose of identifying additional modes of cognitive interaction. In Table 2.6 I synthesize these characteristics, and provide definitions that are based on HRI literature but modified to apply to agent-agent interaction,

including human-human interaction, robot-robot interaction, and human-robot interactions. Since many of these characteristics are interdependent, I also decompose the descriptions where appropriate to form a set of independent modes of cognitive interaction.

This decomposition analysis yields two additional modes of cognitive interaction:

- **Inherent Lag (IL):** characterizes lag in Information Exchange between agents
- **Command Specification (CS):** characterizes the level of functional detail of commands that an agent requires from other agents in order to operate

Synthesized modes of cognitive interaction

I propose a set of five independent modes of cognitive interaction derived from various sources addressing interaction with automation, metrics for human-robotic interaction, and studies on human-robotic interaction. These modes formulate an option space available for architecting the cognitive interaction among agents in a human-robotic system. I define these modes as a property of the cognitive interaction between two agents:

- **Information Exchange (IE):** characterizes the flow of information between two agents in terms of agent requests and transfer of input
- **Information Assessment (IA):** characterizes what state and environment information an agent is able to gather and assess as nominal (according to plan) vs. off-nominal (not according to plan) about other agents.
- **Decision and Action Selection (DS):** characterizes how two agents work (or do not work) together to make execution decisions. ([34, 8] propose taxonomies for this mode of interaction)
- **Inherent Lag (IL):** characterizes lag in Information Exchange between agents
- **Command Specification (CS):** characterizes the level of functional detail of commands that an agent requires from other agents in order to operate

Characteristic	Description	Decomposition
Regulation of Control [31]	Regulation of control is one description of the flow of information between agents and relates to which agent requests assistance or input [31].	Information Exchange: flow of information between two agents
Situational Awareness [31, 69]	Situational awareness is the knowledge of what is going on around you [69]. This also relates to how agents in a human-robotic system make decisions; does one agent make all execution decisions based on multiple, simultaneous task demands or, at the other end of the spectrum, is one agent unaware of all execution decisions [31]?	Information Exchange: flow of information between two agents
		Decision and Action Selection: how two agents work together to make decisions (unilaterally vs collaboratively?)
Communication Latency [31]	Communication latency is the measure of the time lag for communications between agents[31].	Inherent lag: delay in information exchange
Neglect Tolerance [20]	Neglect tolerance is defined as the time between required attentions to an agent [20].	Information Assessment: what state and environment information an agent is able to gather and assess (nominal vs. off-nominal) about other agents.
		Information Exchange: flow of information between two agents
		Decision and Action Selection: how two agents work together to make decisions (unilaterally vs collaboratively?)
Interaction Effort [20]	Interaction effort is a measure of how much attention an agent is demanding [20]	Command Specification: level of functional detail of commands that an agent requires from another agents in order to operate
		Information Assessment: what state and environment information an agent is able to gather and assess (nominal vs. off-nominal) about other agents.
		Decision and Action Selection: how two agents work together to make decisions (unilaterally vs collaboratively?)
Authority Relationships [12]	Authority relationships specify which agent commands the functions to be performed and the level of detail of commands [12]. Examples of authority relationships are: supervisor, operator, peer/collaborator, and bystander [12].	Decision and Action Selection: how two agents work together to make decisions (unilaterally vs collaboratively?)
		Command Specification: level of functional detail of commands that an agent requires from another agents in order to operate

Table 2.6: Review of characteristics of cognitive interaction from HRI sources

Physical interaction

The capacity for physical interaction among agents is an important consideration in architecting human-robotic systems. I present characteristics of physical interaction that I have synthesized from HRI sources. My objective is to generalize these characteristics of physical HRI to formulate an option space for architecting the physical interaction of agents in a human-robotic system.

The DARPA/NSF Interdisciplinary Study on Human-Robot Interaction discusses spatial relationships as a basic taxonomy of HRI [12]. Spatial relationships refer to the viewpoint and intimacy of the operator in relation to the robot [12]. From this taxonomy, intimacy directly relates to the capacity for physical interaction between a human and robot. The study characterizes the intimacy of the operator as: remote, beside, "robo-immersion" or robot's eye, inside [12].

While taxonomies are an interesting analysis tool, I seek to formulate an option space that allows us to design for different classifications of intimacy. I draw from the DARPA/NSF Interdisciplinary Study [12], and papers exploring common metrics for HRI [31] and characteristics of human robotic systems [74] to form a set of generalized, independent physical HRI modes which can be used to describe agent-agent interactions, including human-human interaction, robotic-robotic interactions, and human-robotic interactions. These modes formulate an option space available for architecting the physical interaction among agents in a human-robotic system. I define these modes as a property of the physical interaction between two agents:

- **Response Time (RT):** the time required for one agent to physically intervene with another agent in need of unplanned assistance
- **Availability (AV):** the fraction of time that one agent can devote to physically intervening with another agent in need of unplanned assistance
- **Proximity of Physical Interaction (PPI):** a measure of the proximity of close physical interaction between two agents during nominal operations
- **Duration of Physical Interaction (DPI):** a measure of the duration of close physical interaction between two agents during nominal operations

2.3.3 Generating team architectures

So far I have discussed considerations relating to generating a team architecture, the second building block in the framework proposed in Figure 2-1. This included a discussion of the considerations related to specifying the range of capabilities and constraints for each type of agent, and the modes in which agents interact. However, I have not yet considered how to use this information to formally represent human-robotic team architectures. In this section, I describe what constitutes a team architecture, and how multiple team architectures can be generated.

Defining a team architecture

A human-robotic team architecture consists of all human and robotic agents who participate in fulfilling the objectives of the human-robotic system. For space exploration tasks, human agents include all space and ground personnel participating in any aspect of the operation of the human-robotic system. Similarly, robotic agents include all space- and ground-based robotic and/or automated technologies participating in any aspect of the operation of the human-robotic system.

I define a team architecture by specifying the number of human agents, number of robotic agents, the capabilities and constraints associated with each agent, and the modes available to form scripted and unscripted interactions between agents. I specify another team architecture by changing any of these parameters. As a result, one is able to generate small variations in a human-robotic system such as changing whether a human agent can repair a robotic agent (this corresponds to changing the range of physical interaction between the human and robotic agent). One is also able to generate vastly different human-robotic systems by changing the number of agents and capabilities and constraints associated with each agent.

2.4 Task allocation

In this section, I discuss considerations related to the task allocation, the third building block in the framework we proposed in Figure 2-1. In specifying the capabilities

and constraints of agents, number of agents, and range of interactions among agents as described in the previous section, we form a description of a human-robotic system architecture that we would like to evaluate. However, it is likely that this description does not dictate a unique way in which the agents work together to execute tasks to complete the goal. For example, the end-effector on the Shuttle Arm may be capable of grappling a satellite, and an astronaut located on the end of the Arm may be capable of grappling a satellite. Which agent should perform this task?

In this section, I address the question of which agents should be allocated which actions in order to carry out tasks and achieve the global goal. First I outline the desirable characteristics of a task allocation method for space exploration applications. I then review different methods proposed in literature and suggest possible methods for future work.

2.4.1 Characteristics of task allocation method

The goal of task allocation is to enumerate and provide a framework for scoring different ways in which a human-robotic system can perform a task or set of tasks. Since during the design stage we are not able to directly measure the performance of a fielded system, we evaluate an estimate of performance, or utility of the system. In estimating the utility of the system, we assume that we are capable of assessing the value or cost for each agent to execute an action [36]. Metrics which can be used to form utility measures relevant to space exploration systems and tasks are discussed in detail in Section 2.5. However, we require that our task allocation method provide a framework for clearly defining the utility measure to be evaluated so that we know under what circumstances and assumptions a human-robotic system is optimal. For example, we may allocate actions to agents in the team with the purpose of reducing the operating costs of the system. Or, we may allocate actions to agents based on which agent will perform the job most reliably. It is likely that we would allocate tasks for a team architecture differently depending on whether our primary aim is to optimize our system for cost or reliability. In fact, design often involves selecting and trading-off between multiple metrics.

The task allocation method must also be capable of capturing interrelated utilities or situations in which an agent's utility is dependent on which other tasks or actions other agents execute [36]. For example, the task allocation method must be able to distinguish that the effectiveness (as a measure of utility) of an astronaut grappling a satellite may depend on whether other astronauts or robotic agents are also helping to grapple the satellite or not. The task allocation method must also be capable of imposing task constraints, such as the sequential execution of tasks. For example, one must capture a satellite before repairing it.

2.4.2 Literature review of task allocation methods

The Fitts' list approach [70] is one of the earliest proposed methods for the analysis of task allocation. The Fitts list consists of a table enumerating what "men are better at" and what "machines are better at" and is presented in Table 2.7. However, it is a qualitative approach subject to interpretation [70] and it does not allow the analysis of interaction among subtasks [37].

Men Are Better At	Machines Are Better At
Detecting small amounts of visual auditory, or chemical energy	Responding quickly to control signals
Perceiving patterns of light or sound	Applying great force smoothly and precisely
Improvising and using flexible procedures	Storing information briefly, erasing it completely
Storing information for long periods of time and recalling appropriate parts	Reasoning deductively
Reasoning inductively	
Exercising judgement	

Table 2.7: Fitts list [37]

As an improvement to this method, Ghosh et al. [37] propose a systems approach to task allocation for manufacturing applications which allows the analysis of issues unique to each problem. First, Ghosh et al. form an inventory of the tasks to be performed. Then they discuss the required performance characteristics for performing

the tasks. Finally, they compare capabilities of humans and robots to perform the subtasks [37]. However, this method falls short of providing a framework for incorporating a utility measure to translate the comparison analysis into an evaluation of task allocations.

More recently, Gerkey et al. [36] have proposed a taxonomy of multi-robot task allocation (MRTA) problems and describe how many types of these problems can be related to other, well-studied optimization problems. In these cases, the utility is explicitly incorporated into the method. However, the Gerkey et al. taxonomy of optimization problems shares many of the same fundamental problems as the Fitts and Ghosh et al. approaches; these task allocation methods are not capable of incorporating interrelated utilities or task constraints [36].

I expect to draw methods for our application from task allocation methods currently being developed in the field of artificial intelligence (AI) planning and scheduling. AI planning and scheduling research is working towards providing methods that address complex mixtures of action choices, ordering constraints, and metric quantities [61]. Nearly all practical planning systems have made use of a Hierarchical Task Network (HTN) planning technique [72], and this method is also particularly applicable to our framework. In this method, high-level tasks are reduced down to primitive tasks (or actions) while taking into account ordering constraints to form a network of all feasible actions to complete the task [72]. Time and metrics quantities are easily introduced into this method as well [72]. In HTN planning, the user must specify the combinations of actions that should be used for particular purposes. The interested reader is referred to [29, 28, 45] for a more thorough discussion of HTN planning.

Effinger et al. [27] provide an interesting example for using a type of HTN to describe the Lunar Roving Vehicle (LRV) deployment sequence as documented for Apollo 15. Specifically, they use the Temporal Plan Network (TPN) shown in Figure 2-3 below. This network description incorporates action choices, ordering constraints, and metric quantities.

In this example, the actions "Remove Insulation Blanket" and "Remove Operating

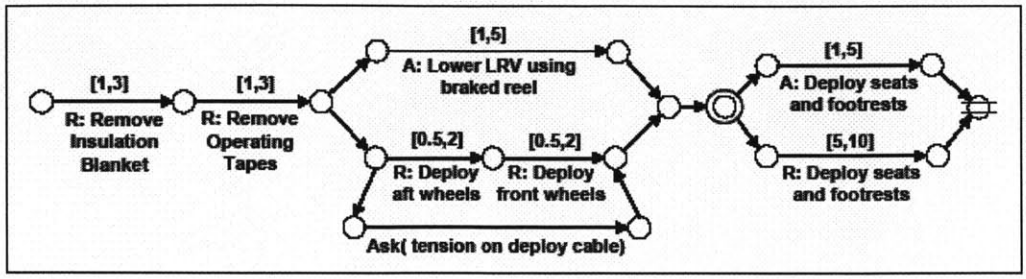

Figure 2-3: Temporal plan network of the Apollo 15 LRV deployment sequence [27]

Tapes" must happen sequentially. However, the actions "Lower LRV using braked wheel" and "Deploy aft wheels" may happen in parallel. The double circle in the diagram indicates a choice node. In this example, the system has a choice of two different ways it can perform the action "Deploy seats and footrests." The metric quantity of time required to perform an action is indicated in brackets above each link in the network.

In the proposed framework, we explicitly specify the subtasks to be performed with ordering constraints in the task specification. In defining a team architecture, we explicitly specify the different combinations of feasible agent actions that may achieve each subtask. Therefore, HTN planning provides a natural method of translating task specification and team architecture definition into a task allocation and metric evaluation framework.

The primary difference between AI planning and scheduling applications and this application is that AI research is focused on solving problems in real-time operations (see [81, 41] for HTN-based examples) whereas we would implement the proposed framework for designing human-robotic systems before any system has been fielded. However, this difference does not limit the application of AI methods such as HTN for the framework.

2.5 Common metrics to compare different human-robotic systems

In this section, I discuss considerations related to evaluating human-robotic system against common, task-based metrics; this corresponds to the fourth building block in the framework I proposed in Figure 2-1. Architecting human-robotic systems for optimal space exploration implies a set of metrics the designer uses to guide optimization. I propose five metrics that are particularly relevant to evaluating the performance of human-robotic systems conducting space exploration tasks: productivity/effectiveness, reliability of successfully completing the task(s), risk to humans in the system, resources required to support system design, implementation, and operations, and flexibility/robustness of the system to changes in task functions or environment states. In this section, I discuss each of these metrics as well as the trade-offs among them.

2.5.1 Productivity/Effectiveness

The productivity or effectiveness of a human-robotic system relates to a measure of quality in performing a task or set of tasks under specific conditions. The purpose of a productivity/effectiveness metric is to capture relevant trade-offs in the quality of task performance for different human-robotic systems and task allocations. The definition for a measure of productivity or effectiveness is unique to the task or set of tasks being performed. For example, if the task is to traverse a planetary surface, we may score the time to traverse as a measure of productivity or effectiveness.

Although the quality of task performance is a major driver in comparing human-robotic system for space exploration tasks, this metric alone does not provide a complete picture of the system performance. While a decision maker may find a low traverse time appealing, he or she will likely want to weigh this advantage against other metrics. For example, the decision maker is likely to prefer a "slower" system-low traverse time-but that is capable of handling a rough or unknown terrain or than

poses lower risk to humans.

2.5.2 Reliability

The reliability of a human-robotic system is the probability that the human-robotic system will perform its required task during a specified interval of time under specific conditions. In other words, assuming there is a deterministically specified task to be performed and given an environment in which the task is to be performed, reliability is the probability that the human-robotic system will successfully complete the task in the time allotted. Although there is a dearth of resources discussing methods for analyzing the reliability of human-robotic systems, the reader is referred to [15, 16, 22, 23, 55, 68, 79] for methods of analyzing the reliability of robotic systems.

The description of team architecture specifies causal relations for how agents in the human-robotic system carry out actions to complete the task or set of tasks. However, there are uncertainties in these causal relations. For example, in the Intelsat-VI repair task there is some amount of uncertainty as to whether the EVA astronaut will be able to secure himself to the RMS and whether the RMS will function as expected in maneuvering the astronaut. These causal relationships and associated uncertainties have direct implications for the reliability of the system. It is important to note that these uncertainties are human-robotic system specific, meaning they are applicable to a specific human-robotic system architecture. They are not related to the system-independent task and environmental uncertainties described in the task specification in Table 2.5. In order to quantify reliability, we must assume a specific task and environment, without considering the implications any of the uncertainties enumerated in the task specification. (I discuss the relationship between metrics and uncertainty in task specification in Section 2.5.5.) The reliability metric of human-robotic system quantifies the relevant uncertainties in how each human-robotic system carries out the actions (as dictated by a particular task allocation) to complete the task or set of tasks.

2.5.3 Risk to humans

A measure of the risk to humans when carrying out a task or set of tasks is a particularly important consideration in evaluating human-robotic systems for space exploration applications. For each human-robotic system -task allocation combination, the designer must analyze and score the extent to which potential undesirable consequences threaten the well-being of humans in the human-robotic system. For example, a human-robotic system task allocation that requires astronauts to perform multiple EVAs is inherently more risky to the astronauts than a human-robotic system task allocation in which humans tele-operate a robotic agent from Earth. As a second example, a human-robotic system task allocation that requires human and robotic agents to work in close physical proximity is inherently more risky to the astronauts than if the astronaut operations are physically segregated from robotic operations. As with the reliability metric, risk to humans is a property of a specific task allocation for a human-robotic system and is evaluated without considering the implications any of the uncertainties enumerated in the task specification. The reader is referred to [22, 2, 30, 82] for methods of analyzing the risk that a human-robotic system poses to humans.

2.5.4 Resources

The resources required to support the operations of a human-robotic system are also a major driver in comparing and selecting a system for space exploration applications. A resource metric captures the time, money, and other resources required to design, implement, and operate the human-robotic system. This includes considerations such as the number of ground and space-based support personnel required during the nominal operation of the human-robotic system, the duration of operations support, supporting ground infrastructure, and space-based infrastructure. The number of ground and space-based support personnel includes those required to operate, monitor, and communicate with robotic agents, monitor critical subsystems supporting manned activities, conduct task-level planning and scheduling, and other activities. The du-

ration of operations support is a key resource driver; will the ground and space-based personnel support operations for one week or one year? Ground-based infrastructure requirements include accommodations for ground support and control of the human-robotic system. For example, ground support and control may be able to make use of existing structures during the operations of one human-robotic system architecture, while a different human-robotic system architecture requires a specially-designed facility. Finally, the infrastructure required in space for operation of the human-robotic system must also be included in a resource metric. Considerations such as the quantity and mass of payloads to be launched, type of launch vehicle required (man-rated or cargo), destination of payload, and type of payload are all relevant to addressing the infrastructure required in space.

2.5.5 Flexibility/Robustness

The nature of space exploration requires that human-robotic systems perform tasks in changing environments that are difficult to fully characterize, and the capability for flexibility and robustness are key drivers in evaluating a human-robotic system. Flexibility of a system is defined as "the ability of a design to satisfy changing requirements after the system has been fielded [66]," and robustness of a system is defined as "the ability to satisfy a fixed set of requirements despite changes in the system's environment or within the system itself [66]." The reader is referred to [64, 65, 11, 63] for further discussion of flexibility and robustness of space systems.

In the proposed framework, both flexibility and robustness relate to the ability of the human-robotic system to respond to the uncertainties defined in task specification and environment (see Figure 2-1). Specifically, flexibility is defined as the ability of the human-robotic system to respond to *Subtask Uncertainty*, and robustness is defined as the ability of the human-robotic system to respond to *Environment Uncertainty*.

Of the four metrics discussed, three are defined assuming a specific task or set of tasks, with no uncertainty in the tasks to be performed or the environment in which they are performed. These metrics are: productivity/effectiveness, reliability, and risk to humans. A metric evaluating the flexibility and robustness of differ-

ent human-robotic systems is required to capture how these three metrics change with task uncertainty. For example, the productivity/effectiveness and reliability of human-robotic systems traversing a planetary surface may change drastically depending on whether the scout is navigating a smooth or boulder-strewn field. If mission planners are not certain what type of field the system will be required to explore, then they may prefer to implement a human-robotic system which sacrifices productivity/effectiveness but maintains a relatively high reliability of successfully carrying out the task in an uncertain environment.

2.5.6 Trade-offs among metrics

I propose that the five metrics described in this section capture many of the relevant trade-offs in designing human-robotic systems for space exploration. In capturing these relevant trade-offs, I hypothesize that these metrics will provide a fair and balanced basis for comparing different human-robotic systems and will not skew results towards consistently recommending predominantly human or predominantly robotic systems for space exploration tasks.

The metric capturing productivity/effectiveness of a human-robotic system is not inherently biased towards the abilities of either human or robotic agents. Human agents will score higher on certain tasks and subtasks, and robotic agents will score higher on others. Reliability is a metric equally applicable to both human and robotic agents as well; it is not clear how a human-robotic system will score before carrying through the analysis for a specific task and task allocation. Risk to humans and resources required to support operations of a human-robotic system are two critically important metrics to decision makers. However, they tend to drive the solution towards predominantly robotic systems without capturing the true value that humans bring to space flight - the ability to accommodate uncertainty. The flexibility and robustness of a human-robotic system are necessary and often forgotten performance metrics required to provide a balanced comparison of different human-robotic systems.

2.5.7 Towards evaluation of metrics

While a discussion of methods for evaluating these high-level, common metrics is beyond the scope of this thesis, it is important to recognize that evaluation of these metrics may require an experimental-based understanding of lower-level metrics. For example, works analyzing existent HRI systems identify human situational awareness as a major driver in HRI system performance [69, 25, 26]. Human situational awareness have implications for a number of high-level metrics including productivity/effectiveness and risk to humans. Therefore, an experimental-based understanding of lower-level metrics may prove useful in providing analytical relationships between the description of system architecture and common, high-level metrics.

2.6 Summary

In this chapter, I presented a framework for architecting human-robotic system for space exploration tasks (Figure 2-1). I discussed considerations related to each of the building blocks of this framework: 1) specifying tasks, 2) generating human-robotic system architectures, 3) allocating functions to agents in the system, and 4) evaluating the system against common task-based metrics.

In discussing task specification, I outlined the characteristics a description of task and environment should capture. I have drawn on the literature to discuss the different methods of task specification. From this literature review, I synthesized and proposed a method to specify tasks and environment. Finally, I presented an illustrative example of our task specification method.

In addressing considerations related to generating human-robotic system architectures, I have drawn from literature to discuss considerations related to specifying capabilities and constraints for each agent and the modes in which agents interact. I also described how these considerations can be used to generate multiple team architectures.

In the discussion on task allocation, I addressed the question of which agents should be allocated which actions in order to carry out tasks and achieve the global

goal. I outlined the desirable characteristics of a task allocation method for space exploration applications and reviewed different methods proposed in literature. Finally I suggested possible methods for future work.

I also proposed a common set of metrics relevant to comparing different human-robotic systems performing space exploration tasks and discussed trade-offs among these metrics. The next chapter is aimed at further developing a formal representation of human-robotic systems.

Chapter 3

Towards a formal representation of human-robotic systems in space exploration

3.1 Introduction

3.1.1 Motivation: Why do we need a formal team representation?

In this chapter I present work towards formal methods to represent human-robotic system architectures. This work builds on the previous chapter discussing the considerations in architecting human-robotic systems. My objective in developing a formal method for representing teams of humans and robots is to provide a basis for a standard means of evaluating human-robotic systems against a common set of metrics. Evaluating different human-robotic systems based on a non-standard description of system architecture will make it difficult to compare the evaluation results for different systems. The designer must systematically and formally capture the characteristics that describe each human-robotic system to ensure that the evaluations of different human-robotic systems are comparable.

3.1.2 Outline of chapter

In Section 3.2, I discuss the important elements that must be captured in the formal representation of a human-robotic system. In Section 3.3, I illustrate initial work towards formalizing the description of these elements through the case-study of the terrestrial human-robotic Nursebot System, which provides assistance to nursing home residents and caregivers.

3.2 What are the important elements to capture in a team representation and why?

3.2.1 Elements of a human-robotic team architecture

I define a human-robotic team architecture to consist of all human and robotic agents that participate in fulfilling the objectives of the human-robotic system. In the context of space exploration, human agents include all space and ground personnel participating in any aspect of the operation of the human-robotic system. Similarly, robotic agents include all space- and ground-based robotic systems "which have complex, dynamic control systems, exhibit autonomy and cognition, and which operate in a changing real-world environment [69]."

A formal representation of team architecture must specify the number of human agents, number of robotic agents, the capabilities and constraints associated with each agent, and the range of interactions between agents. A change in any of these elements results in a different team architecture. Therefore, the formal representation of human-robotic systems must be capable of capturing variations in a human-robotic system such as changing whether or not a human agent can repair a robotic agent (this corresponds to changing the range of physical interaction between the human and robotic agent). The formal representation must also be able to capture vastly different human-robotic systems through changes in the number of agents and capabilities and constraints associated with each agent.

To capture all of these elements, we require formal methods for decomposing the

system-independent task specification into system-specific functions with ordering and timing constraints (1), as I will further discuss shortly. We also require methods for identifying the interactions necessary to fulfill system-specific functions as well as for specifying the characteristics of these interactions (2). Finally, we require methods for specifying how system-specific components of form are utilized to fulfill system-specific functions (3), and for identifying the design parameters relevant to analyzing the autonomy of the system (4). In this section, I discuss the motivation for developing each of these four methods.

3.2.2 System-specific functions

In the previous chapter I discussed considerations related to specifying tasks in a way that allows the comparison of different human-robotic systems. These high-level tasks describe what the system does in language that is system-independent in order to formulate common metrics to compare the advantages and disadvantages of different human-robotic systems [60]. However, each human-robotic system will likely carry out these tasks and subtasks through different means, or through different system-specific functions. A critical aspect to describing the architecture of a human-robotic system is to formally specify what system-specific functions aggregate to fulfill the system-independent tasks and subtasks. Given a system-independent task specification with ordering constraints, I identify the functions that a specific human-robotic system performs to fulfill each system-independent subtask as shown in Figure 3-1.

I propose that organizing the system-specific functions by system-independent subtasks provides a natural basis for translating the task specification and representation of team-architecture into a task allocation analysis. We require a task allocation analysis as a component in the framework for leveraging the capabilities of humans and robots in space exploration because it is likely that there is no unique way in which to allocate functions to agents in a human-robotic system to fulfill the system-independent task specification. Therefore in describing a team architecture, we must explicitly specify the different combinations of feasible agent actions that may achieve each subtask. The goal of task allocation is to enumerate and provide a framework

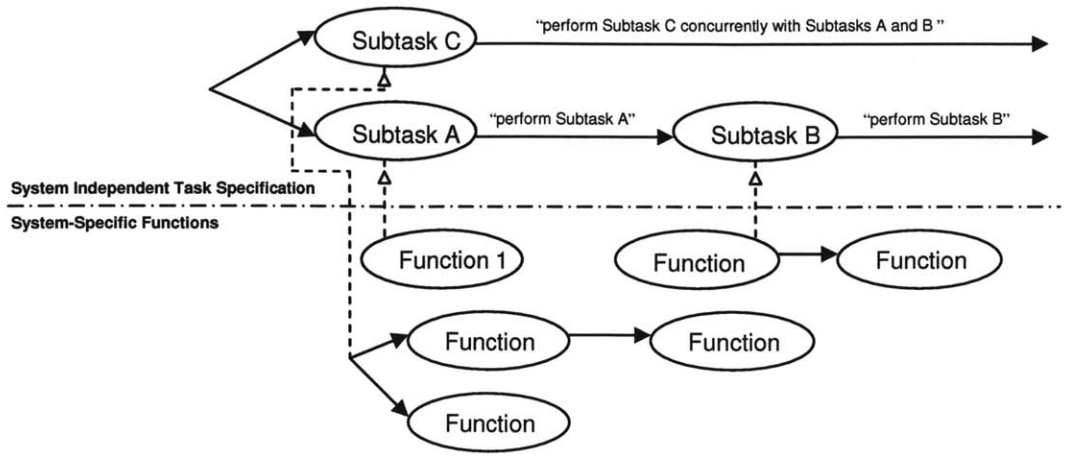

Figure 3-1: Identifying system-specific functions

for scoring all the different ways in which a human-robotic system can perform a task or set of tasks. In the previous chapter I reviewed the desirable characteristics of a task allocation method for this application and identified the Hierarchical Task Network (HTN), an artificial intelligence (AI) planning and scheduling technique, as a promising candidate method. In this method, high-level tasks are reduced down to primitive tasks (or functions) while taking into account ordering (and possibly timing) constraints to form a network of all feasible actions to complete the task [4]. The proposed method for organizing system-specific functions shown in Figure 3-1 captures many of the elements necessary to implement an HTN by specifying the way in which high-level task specification is reduced down to system-specific functions.

Now that I have motivated the mapping of system-independent tasks and subtasks into system-specific functions as shown in Figure 3-1, I will discuss what other descriptive information must be included with the system-specific functions. At least two attributes must be specified with each system-specific function: ordering constraints and timing. Just as we specify ordering constraints among subtasks in the system-independent task specification, it is important to specify ordering constraints among the system-specific functions where appropriate. This provides temporal relations describing how system-specific functions aggregate to fulfill subtasks.

Specifying timing constraints on each system-specific function is also important to

describing the system architecture. While it is conceivable that two different human-robotic system architectures may perform the same functions to fulfill the specified task or set of tasks, the systems may implement the functions in different ways. One way of capturing the implication of different implementations is to describe the characteristic time (or time range) required for the human-robotic system to carry out each function. For example, imagine two autonomous planetary rovers that perform the same function of path planning, but each implement the function using different software algorithms. The first rover system architecture requires milliseconds to path plan, while the second architecture requires seconds. Time to implement a function is in many cases a function of the agent's capabilities and must be captured in the formal representation of the system. However, it is important to note that characteristics of different implementations must be captured in other ways as well, including which agents and what components of agents implement functions. This is addressed in the following sections.

3.2.3 Identifying agent interactions

In specifying the organization and attributes of system-specific functions, I have not yet addressed to what level the functions should be decomposed. For example, it is possible to decompose the rover function of navigating across a planetary surface to the functions performed by the nuts and bolts of actuators and cameras. However, decomposing functions to this level does not add insight into the important elements of the system architecture. In formally describing a team architecture, I propose that system-specific functions be decomposed to a level such that one agent may be assigned to perform each function as illustrated in Figure 3-2 below.

This method serves to highlight which system-specific functions require agent-agent interaction to fulfill a subtask, and the role of each agent in fulfilling the function. For example, in Figure 3-2, the system requires interaction between Agent 1 and Agent 2 to perform Function 1 and also requires interaction between Agent 1 and Agent 2 to perform Function 2.1. Although the same two agents are interacting to perform Function 1 and Function 2.1, it possible that the agents must interact

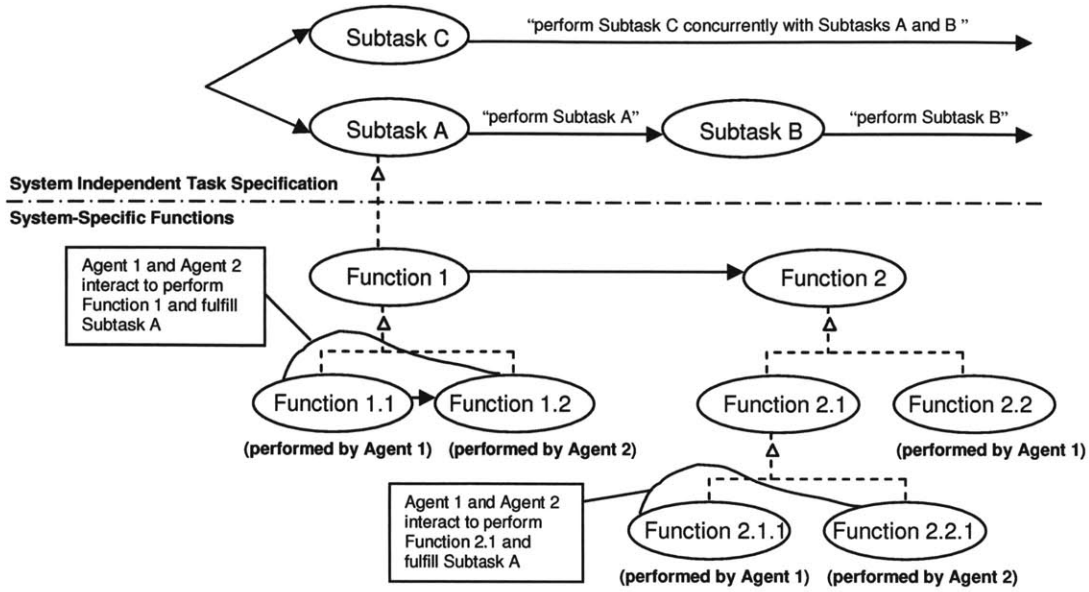

Figure 3-2: Identifying agent interactions

in different ways in order to perform each function. The formal description of system architecture must dictate the specific ways in which the agents interact to fulfill different functions.

Also, this method provides uniform criteria for choosing the level of decomposition of system-specific functions: system-specific functions are decomposed to the level necessary to capture all agent-agent interactions. An example of decomposing functions and identifying interactions is presented in the case study (Section 3.3). Each of these interactions can be further described in term of the modes of cognitive and physical interaction presented in Chapter 1.

3.2.4 Describing the implementation of human-robotic systems

So far I have discussed ways for describing a human-robotic system architecture in terms of system-specific functions and interactions among agents to fulfill these functions. However, an important element of the system architecture is the hardware and software implementation utilized to fulfill system-specific functions. Two systems which carry out the same system-specific functions using different hardware or

software components may exhibit different capabilities. Imagine two human-robotic systems that perform the same system-specific functions, with the same agents, and the same agent interactions. The only difference between the two systems is the technology used by the agents to fulfill the functions; for example, one system uses laser range finders to sense the environment and the other system uses sonar sensors. This simple difference between the two systems may have great implications when comparing the system based on common metrics such as effectiveness, reliability, resources expended, risk to humans, or flexibility/robustness. Therefore, the formal representation of a human-robotic system must include a description of system hardware and software components. In Figure 3-3 below I build on the approach for representing system-specific functions and agent interactions and include a description of the system-specific components of form utilized to fulfill system-specific functions.

In specifying the hardware and software components used to implement system-specific functions, I delineate the components which must be used together to perform the function with an "And" symbol and I delineate options between components with an "Or" symbol. These symbols are included in the representation to capture the fact that many systems maintain redundant components to fulfill a function.

Just as system-specific functions can be decomposed down to an arbitrary level, so can system components. Therefore we require a method for specifying the appropriate level of decomposition for system components in describing the system architecture. The appropriate level to which components must be decomposed is specific to each application. However, I propose a guiding principle: the decomposition of components must capture all relevant differences in form associated with the set of human-robotic systems the designer is comparing. If the designer is comparing vastly different human-robotic systems, all relevant differences in form may be captured at the first level decomposition of components. However, the designer may be required to specify a multi-level decomposition of components if comparing very similar human-robotic systems with only slight variations in technology.

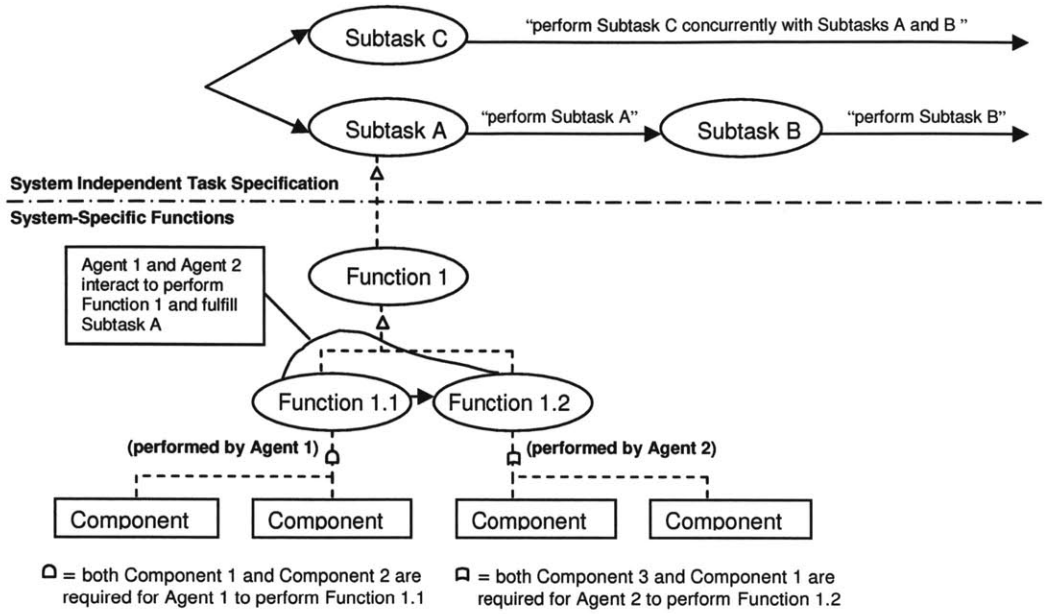

Figure 3-3: Describing the implementation of human-robotic systems

3.2.5 Identifying design parameters for system autonomy

To this point, I have discussed approaches for describing a human-robotic system architecture in terms of system-specific functions, interactions among agents and system-specific components of form to fulfill these functions. However, the representation of a human-robotic system is still missing a full description of the capabilities and constraints of agents. So far it only includes timing constraints describing the characteristic time for a system to fulfill a function. It does not yet capture design parameters such as resource constraints, speed, range of motion, and other capabilities and constraints of the system in performing the functions. In order to specify the design parameters relevant to describing the system, we first require a method to identify the set of capabilities and constraints that are relevant to how agents perform system-specific functions in the context of the operating environment. We seek to avoid specifying capabilities and constraints that do not contribute to our understanding of how the system operates in its environment. For example, we would not specify the speed of an agent if we know that speed has no bearing on the agent's ability to perform its functions within the operating environment.

The nature of space exploration requires that human-robotic systems perform tasks in changing environments, and we capture these uncertainties in the environment in the task specification. These uncertainties or variations are a part of the description of the operating environment. Therefore we are interested in identifying the design factors relevant to how agents perform functions in changing environments, including different environments and transitions between environments. For example, if we identify temperature as a relevant environmental state in a task specification, we may be interested in identifying the design factors relevant to how an agent is able to perform in cold temperatures, warm temperatures, and transitions between cold and warm temperatures. In other words, we are interested in identifying the set of design parameters that affect the system autonomy, or ability of the system to accommodate the range of possible variations in the environment in pursuit of its goals [77].

I propose that the concept of autonomy is a useful tool for identifying relevant design parameters to describe the capabilities and constraints of agents. We can identify relevant design parameters by applying the following question template:

"What design parameters related to performing [**a system-specific function in fulfilling a Subtask**]are important for determining [**agent**]'sability to accommodate variations in [**specific environment state**]?"

Consider the simple human-robotic system described in Figure 3-4.

In this system, the Astronaut tele-operates the Rover in real-time with negligible time delay. The Astronaut remotely provides navigation direction for the Rover and the Rover then actuates motion to traverse across the field according to the navigation directions. I identify relevant design parameters to describe the capabilities and constraints of the Astronaut and Rover by applying the question template and answering the following questions:

1. What design parameters related to **actuating motion for traversing across the boulder-strewn field**are important for determining the **Rover's** ability to accommodate variations in **boulder size**?

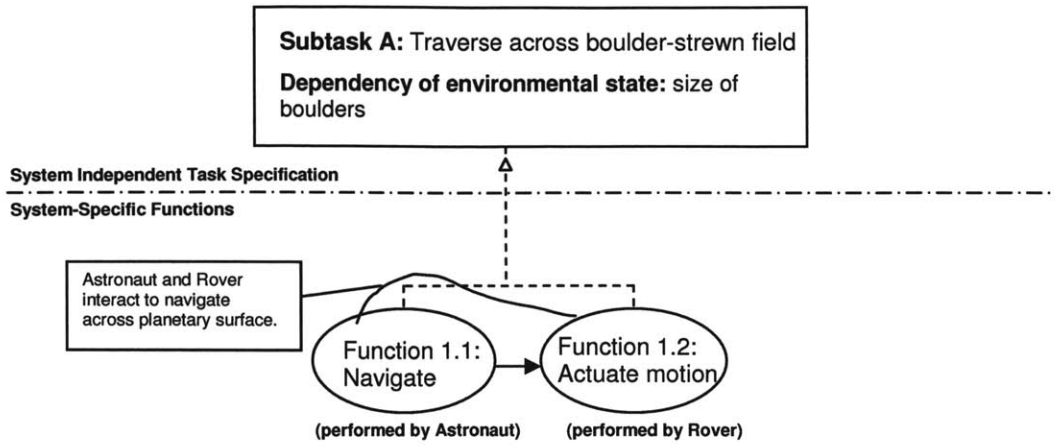

Figure 3-4: Simple human-robotic system

2. What design parameters related to **navigating for traversing across the boulder-strewn field**are important for determining the **Astronaut's** ability to accommodate variations in **boulder size**?

In answering the first question, we determine that the Rover's wheel size and configuration are some of the critical design parameters for determining the Rover's ability to accommodate variations in boulder size. Therefore, we must specify these design parameters to describe the Rover's capabilities and constraints in performing this system-specific function.

We also determine that Astronaut reaction time, ability to gauge distance and speed with information available are some of the critical parameters for determining an Astronaut's ability to accommodate variations in boulder size. (Information available is a function of Information Assessment specified in the description of interaction.) Therefore, we must specify these parameters to describe the Astronaut's capabilities and constraints in performing this system-specific function.

We can apply this question template to each system-specific function performed by an agent to generate a list of the function-specific design parameters. The specification of each of these design parameters then describes the capabilities and constraints of each agent relevant to the operating environment. Although the design parameters that we specify for different human-robotic systems performing the same task are

likely to vary, the common question template used to identify these design parameters provides a standard basis for comparing the performance of different human-robotic systems in terms of flexibility and robustness, the ability to respond to uncertainty in task specification and environment.

3.3 Case Study: Nursebot System

In this section, I illustrate initial work towards formalizing the description of human-robotic systems through the case-study of the terrestrial human-robotic Nursebot System, which was developed to assist nursing home residents with mild and cognitive physical impairments and support nurses in their daily activities [58, 6, 48, 59]. I choose this terrestrial system for a case-study for three reasons. First, unlike many human-robotic systems designed for space applications, the current Nursebot System has few agents: a robotic agent named Nursebot with artificial intelligence capabilities, one nursing home Resident and one Caregiver. The small number of agents in this human-robotic system is an advantage in formulating an illustrative case-study. Second, this system exhibits the current state of the art in human-robot interactions. Third, this system is both well-known and well-documented.

The Nursebot System task specification includes seven subtasks. Nursebot must contact the nursing home resident, then remind the Resident of relevant appointments, and then if necessary accompany the resident to appointments. While carrying out these three subtasks, Nursebot must concurrently maintain an accurate internal model of the Resident's planned daily activities and status. Nursebot is also tasked with providing information of interest to the Resident, and must cease interacting with the Resident upon request [58]. The task specification, including ordering constraints among subtasks, for the Nursebot System is summarized in Figure 3-5.

Each of these subtasks in the task specification has dependencies on environmental states that affect the design and performance of the human-robotic system. One of these environmental states is shown in Figure 3-5; the subtask "Maintain accurate description of Resident's daily activities and status" is dependent on the time-variant

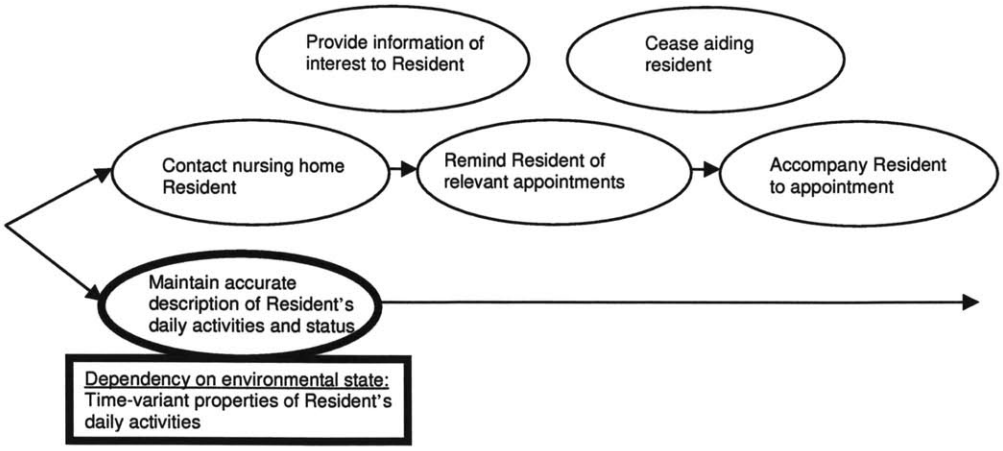

Figure 3-5: Task specification for Nursebot System adapted from [58]

properties of the Resident's daily activities. For example, in a real-world environment the Resident's daily schedule changes often during the day. As a result it is necessary to monitor the Resident during the day to maintain an accurate description of the Resident's daily activities and status. On the other hand, if the Nursebot System is operating in a controlled environment where the Resident's schedule does not change during the day, it may not be necessary to monitor the Resident.

In the following sections, I illustrate some of the formal methods for describing how the Nursebot System fulfills the subtask "Maintain accurate description of Resident's daily activities and status." I present a description of the system-specific functions and ordering constraints, identify and provide an example for describing agent interactions, and identify design parameters for system autonomy. Finally, I discuss how this case-study may be expanded to a complete, formal description of the Nursebot System.

3.3.1 Specifying system-specific functions and ordering constraints

In Figure 3-6 I present a formal description of the system-specific functions that the Nursebot System carries out in order to fulfill the subtask of maintaining an accurate description of the Resident's daily activities and status. Arrows in the figure indicate

the ordering constraints among functions.

In order to maintain an accurate description of the Resident's daily activities and status, the Nursebot System both models a plan of activities and infers the activities performed by the Residents throughout the day. Modeling the plan of activities includes adding new activities, modifying or deleting activities, conducting time-dependent updates, and then propagating modifications and status updates. Inferring the activities performed by the Resident includes confirming compliance reminders, tracking the Resident, reasoning about site-specific tasks, and reasoning about the likelihood a planned activity has been executed [58]. Some of these system-specific functions are decomposed further such that one agent (Nursebot, Resident, or Caregiver) can be assigned to perform each function. This allows the identification of agent interactions as described in the next section.

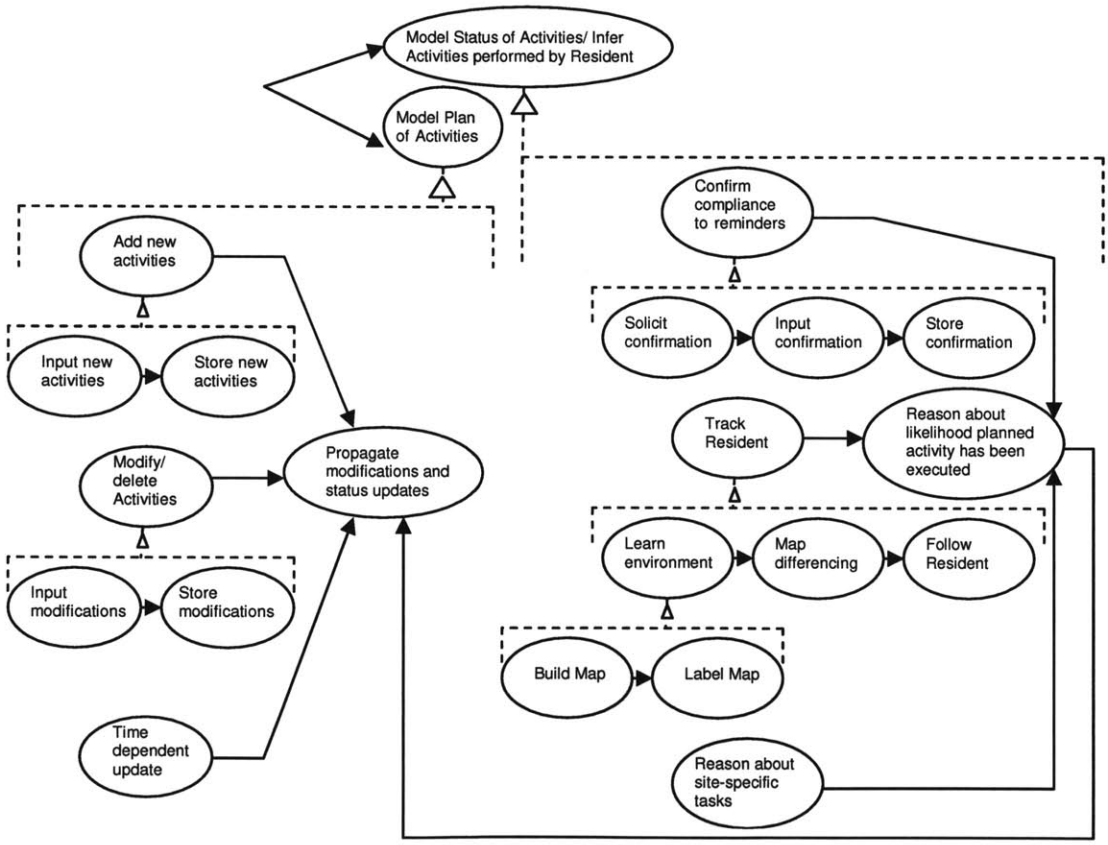

Figure 3-6: Example of system-specific functions for Nursebot System [58]

3.3.2 Identifying and describing agent interactions

In the previous section I have decomposed the system-specific functions to a level such that one agent can be assigned to perform each function and have organized the functions in the diagram so as to highlight ordering constraints. In Figure 3-7 I now re-organize the system-specific functions in the diagram into a tree-structure without ordering constraints to better identify and highlight agent interactions. I then specify which agent performs each function. If more than one agent can perform the function, we use the "Or" symbol. Curves connecting two agents are used to highlight interactions.

I identify that there are nine different interactions among agents that may take place to fulfill the subtask of maintaining an accurate description of the Resident's daily activities and status. In order for the Nursebot System to track the Resident, it must learn the environment. Nursebot is capable of building a map of the nursing home environment; however the Caregiver must then interact with Nursebot to label this map (1 possible interaction, represented in the lower left of Figure 3-7). Either the Caregiver or the Resident can input new activities and/or modifications to the model plan of activities stored by Nursebot (4 possible interactions, represented in the upper half of Figure 3-7). Nursebot can solicit confirmation to reminders either from the Resident or Caregiver (2 possible interactions). And either the Caregiver or Resident can issue confirmation to Nursebot's reminders (2 possible interactions, represented in the lower right of Figure 3-7).

Each of these interactions can be described by a set of physical and cognitive characteristics. For example, consider the interaction between the Caregiver and Nursebot while adding new activities to the model. The physical characteristics to describe this interaction include: Proximity of Physical Interaction (PPI), Duration of Physical Interaction (DPI), Response Time (RT), and Availability (AV). The Caregiver and Nursebot physically contact while adding new activities to the model plan (PPI). The duration of physical interaction each time the Caregiver adds a new activity to the Nursebot's model is on the order of one minute (DPI). The response

time required for the Caregiver to physically intervene when Nursebot is in need of assistance is on the order of one minute; Nursebot is not able to physically intervene if the Caregiver needs unplanned assistance (RT). Finally, there is no constraint on the amount of time the Caregiver can devote to physically intervening when Nursebot is in need of assistance; Nursebot is not able to devote any time to physically intervening with the Caregiver (AV).

The cognitive characteristics to describe this interaction include: Information Exchange (IE), Information Assessment (IA), Decision and Action Selection (DS), Information Lag (IL), and Command Specification (CS). Information Exchange characterizes the flow of information between the Caregiver and Nursebot in terms of agent requests and transfer of input. The Caregiver can request the addition of new activities, whereas the Nursebot cannot do this; information about activities in the form of text is transferred from the Caregiver to Nursebot, and Nursebot provides visual confirmation that the information has been stored (IE). The Caregiver, who does not have expert knowledge about Nursebot, is able to detect but likely not identify faults of the Nursebot components related to adding new activities to the model plan; Nursebot is not able to sense state or environment information about the Caregiver (IA). The Caregiver unilaterally decides what activities need to be input in the model plan (DS). The time delay in information exchange between the Caregiver and Nursebot is negligible (IL). Finally, Nursebot requires the Caregiver to input detailed description for the new activity including constraints or preferences regarding the time or manner of their performance (CS).

3.3.3 Identifying design parameters for system autonomy

I now identify and specify the design parameters for describing the capabilities and constraints of the Nursebot System that are relevant to how agents perform system-specific functions in the context of the operating environment. To illustrate this method I apply the question template described in Section 3.2.5 to the tracking-related functions that the Nursebot agent performs. These system-specific functions include building a map of the nursing home environment, map-differencing to find

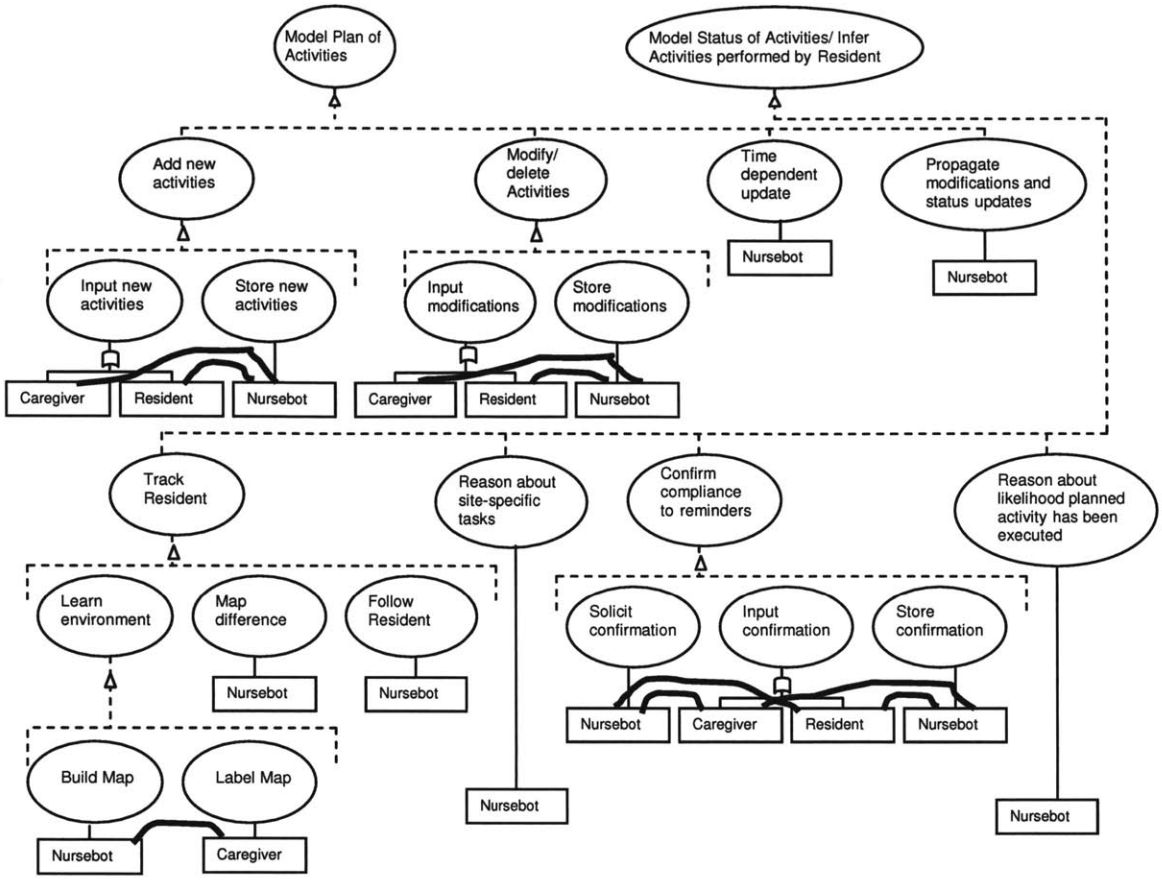

Figure 3-7: Example of identifying agent interactions for Nursebot System

people (to avoid or follow), and following the Resident. I answer the following three questions to identify the design parameters relevant to these three functions:

1. What design parameters related to [**building a map to track the Resident and infer activities performed by the Resident**]are important for determining [**Nursebot**]**'s** ability to accommodate variations in [**the time-variant properties of the Resident's daily activities**]?

   If we are not able to fully predict the Resident's daily activities and Nursebot must track the Resident, this raises the issue: what is the fidelity of Nursebot's model of the environment? For example, in following the Resident to unscheduled locations, could Nursebot find itself in a location it does not recognize on its map? To answer these questions I specify the following design parameters:

   - Mapping environment (indoors/outdoors, floor surface, etc)

- Map dimension (2-D, 3-D, etc.)
- Map resolution/grid size
- Estimated state information of Nursebot
  - Precision and accuracy of location (x-, y-)
  - Precision and accuracy of orientation
- Sensor specifications
  - Field of view
  - Detection capabilities, limitations

2. What design parameters related to [**map differencing to track the Resident to infer activities performed by the Resident**]are important for determining [**Nursebot**]'s ability to accommodate variations in [**the time-variant properties of the Resident's daily activities**]?

   Nursebot identifies people and tracks the Resident by comparing a map of the environment generated in real-time to the map it has learned previously. In order to determine how well Nursebot can track the Resident, we need to determine the fidelity of the map generated in real-time. We also need to understand how Nursebot identifies the Resident (among other people) for tracking. Finally we need understand the capabilities and constraints of the algorithms used in tracking the Resident. To address these issues I specify the following design parameters:

   - Map dimension (2-D, 3-D, etc)
   - Map resolution/grid size
   - Sensor specifications
     - Field of view
     - Detection capabilities, limitations
   - Criteria for resident identification (initial location, face identification, etc.)
   - Estimated state information of people, including Resident

- Precision and accuracy of estimated location (x-, y-) for Nursebot
- Precision and accuracy of estimated Nursebot orientation
- Precision and accuracy of estimated location (x-, y-) of people/Resident relative to Nursebot
- Parameters for probabilistic model of people's motion

3. What design parameters related to [**following the Resident to infer activities performed by the Resident**]are important for determining [**Nursebot**]**'s** ability to accommodate variations in [**the time-variant properties of the Resident's daily activities**]?

   Once Nursebot has built a map of the environment and is able to map difference to determine the position of the Resident, Nursebot must follow the Resident through his or her daily activities. We must specify the following design parameters to understand the capabilities and constraints of Nursebot in following the Resident to both scheduled and unscheduled activities:

   - Target following distance
   - Speed range
   - Time constant for matching Resident's speed
   - Parameters for mobile and stationary obstacle avoidance algorithm

3.3.4 Expanding the case-study to a complete, formal description of Nursebot System architecture

In this illustrative case-study I have considered one of the subtasks performed by the Nursebot System and presented a description of the system-specific functions and ordering constraints which fulfill the subtask, identified the interactions to fulfill the subtask, provided an example for specifying the characteristics of these interactions, and identified the design parameters relevant to Nursebot's functions in tracking the Resident. To extend this illustrative case-study to a complete, formal description of the Nursebot System we would have to:

- Describe the system-specific functions, ordering constraints, and timing for each of the six subtasks in the task specification presented in Figure 3-5. (Note that the case-study did not include a description of timing.)
- Identify the interactions to fulfill all six subtasks and specify the characteristics of each of these interactions.
- Describe the implementation of the human-robotic system as presented in Section 3.2.4. (Note that the case-study did not include a description of implementation.)
- Identify and specify the design factors for system autonomy for each system-specific function and associated subtask environmental states.

3.4 Summary

In this chapter I discussed methods for formally describing human-robotic systems (building block 2 of Figure 2-1). This includes methods for: decomposing the system-independent task specification into system-specific functions with ordering and timing constraints (1); identifying the interactions necessary to fulfill system-specific functions as well as for specifying the characteristics of these interactions (2); specifying how system-specific components of form are utilized to fulfill system-specific functions (3); and identifying the design parameters relevant to analyzing the autonomy of the system (4). My objective in developing these formal methods for representing teams of humans and robots is to provide a basis for a standard means of evaluating human-robotic systems against a common set of metrics. In the next chapter, I begin to investiage an analytical basis for evaluating human-robotic systems in terms of high-level metrics.

Chapter 4

Analytical basis for evaluating the effect of unplanned interventions on the effectiveness of a human-robot system

4.1 Introduction and motivation

Human-robot systems are being increasingly considered, and used in a number of military operations, civilian search and rescue operations, and are proposed as an integral part of future space missions to the Moon and Mars [14, 12, 67]. The increased relevance of human-robot systems raises the issue of how to optimally (and reliably) design these systems to best leverage the varied capabilities of humans and robots. The question of optimality in turn raises the question of what metrics to use in order to guide the design, and evaluate the performance, of human-robot systems. Unfortunately, an analytical framework of common metrics does not currently exist to compare the performance of different human-robot systems. Formulating such a framework is challenging in part because techniques do not exist to incorporate

[1]This chapter is based on work under consideration for publication with Reliability Engineering and System Safety.

the effect of human-robot interaction into methods for evaluating metrics such as productivity, reliability, and risk to humans. The operational environment of human-robot systems is often hostile to the human agents and the metrics of productivity, reliability, and risk to humans are strongly coupled.

This chapter addresses the relationships between productivity, reliability, and risk to humans for human-robot systems operating in a hostile environment. Objectives for maximizing the effectiveness of a human-robot system are presented which capture these coupled relationships, and parameters are proposed to characterize unplanned interventions between a human and robot (Section 4.2). The effect of unplanned interventions on measures of effectiveness is discussed qualitatively (Section 4.3). Next, the effect of unplanned interventions is formulated analytically in terms of the reliability parameters defined by the author (Section 4.4). The potential implications of this preliminary analysis on the design and evaluation of human-robot systems are then discussed (Section 4.5).

4.2 Metrics and definitions

In designing human-robot systems to operate in a hostile environment, two objectives ought to be considered:

1. To maximize the amount of useful work that human agents can do within a time window (human productivity)

2. To minimize the risk to human agents. In many situations, this entails minimizing time human agents spend in the hostile environment (exposure).

The details of the specific task, the mission-level timeline, and the environment in which the task is to be performed determine which of these measures is most appropriate. Minimizing exposure may be the primary objective in many terrestrial applications involving few tasks, dangerous and unpredictable environments, and low penalties for entering and exiting the operational environment. For example, urban search and rescue teams may minimize the risk to rescuers by using robotic agents to

search buildings in danger of structural collapse and identify victims before sending in rescuers, even though this strategy may not maximize the productivity of each rescuer. In contrast, maximizing human productivity may be the primary objective in situations where there is some penalty for entering and exiting the operating environment. For example, astronauts preparing for an extravehicular activity (EVA) on the Space Shuttle and International Space Station are required to breathe pure oxygen for up to a few hours before the EVA to avoid decompression sickness. With this penalty, mission planners may prefer to maximize the astronauts' productivity during the span of a single EVA rather than require the astronauts to perform many short EVAs - even if many short EVAs would minimize the time astronauts spend in the hostile environment. While both maximizing human productivity and minimizing exposure are important considerations for human space flight, maximizing human productivity is also likely to be the dominant objective for situations in which astronauts have other useful work to do in the extravehicular environment. For example, robotic agents may be deployed to reduce the amount of time humans would spend working on a particular task. Astronauts would then use the extra time to begin working on other tasks, thereby increasing the human productivity during an EVA. The motivation for the following discussions is to formulate an analytical basis for investigating the effect that human-robot interaction has on the objectives to maximize human productivity and minimize exposure.

This preliminary analysis investigates the effect of unplanned interventions, a specific type of human-robot interaction, on each of these objectives. The following definitions are presented for the purpose of this analysis:

An **intervention** is defined as a robotic agent receiving unplanned assistance from a human agent.

Mean Time Between Interventions (MTBI)is the mean time that a human-robot system operates nominally (human and robotic agents are not engaging in an intervention). This is defined as:

$$MTBI = \frac{t_{btw-int}}{n_{int}} \tag{4.1}$$

where $t_{btw-int}$ is the cumulative time that the system is not engaging in unplanned interventions requiring a human agent, and n_{int} is the number of unplanned interventions requiring a human agent.

Mean Time Between Interventions (MTBI) is analogous to the Mean Time Between Failures (MTBF) as defined in the IEEE Standard [56][1]. However, while MTBF refers to component or system reliabilities, MTBI takes on a broader meaning. MTBI is a function of:

1. The environment (and uncertainty in the environment) that the system or robotic agent is operating in.

2. The autonomy of the system or robotic agent, in this case defined as the ability to accommodate variations in the environment in pursuit of its goals [77] (in this case, without human intervention).

3. The inherent component or system reliabilities.

For example, the MTBI for a rover operating in a boulder-strewn field may be dependant on the size and distribution of boulders, and also the rover's ability to autonomously navigate among boulders. The inherent reliabilities of the components utilized by the rover while navigating will also affect MTBI. In defining MTBI I have renegotiated the meaning of MTBF to better capture properties of a human-robot system in the same spirit that [43] defined a maintenance free operating period (MFOP) to better analyze and predict properties of aerospace systems.

Mean Time To Intervene (MTTI) is mean duration of interventions. This is defined as:

$$MTTI = \frac{t_{int}}{n_{int}} \tag{4.2}$$

[1]Mean Time Between Failures (MTBF) is defined in IEEE Std. 493-1997 as: The mean exposure time between consecutive failures of a component. It can be estimated by dividing the exposure time by the number of failures in that period, provided that a sufficient number of failures has occurred in that period.

where t_{int} is the cumulative time spent by human agents engaging in unplanned interventions and n_{int} is the number of interventions requiring a human agent.

Mean Time To Intervention (MTTI) is analogous to the Mean Time To Repair (MTTR) as defined in the IEEE Standard [56][2]. MTTI is a function of many variables, including (but not limited to):

- the nature of the failure or problem requiring an intervention
- the design parameters describing cognitive abilities and interaction among agents including:
  - the amount of information agents are able to gather about the nature of the failure prior to and during the intervention
  - the amount and type of information that can be transferred between a robotic and human agent.
- the physical distance between a robotic and human agent
- the lag in communications between a robotic agent and human agent
- the available resources and tools

The goal of this work is to build on these objectives and parameters to analytically describe the effect of interventions on the effectiveness of human-robot systems, and explore potential implications for the design and evaluation of human-robot systems. Next, the effect of interventions on system effectiveness is discussed qualitatively to form the basis for an analytical discussion.

[2]Mean Time To Repair (MTTR) is defined in IEEE Std. 493-1997 as: The mean time to repair or replace a failed component. It can be estimated by dividing the summation of repair times by the number of repairs, and, therefore, it is practically the average repair time.

4.3 Qualitative discussion of the effect of interventions

In this section, the concept of a mission timeline is used to qualitatively discuss the effect of interventions. A nominal mission timeline in which the human-robot system performs a collaborative task without unplanned interventions is shown in Figure 4-1a. The mission time, $t_{max.mission}$, is the maximum time window that the human-robot system has to perform a specific task, and represents time constraints associated with humans working in a hostile environment. Consider for example an astronaut performing a spacewalk or a scuba diver on a dive. In these cases the maximum time window is dictated by the amount of life support consumables (e.g. oxygen) the human agents can carry with them. This is the same maximum time window referred to in the objective to maximize human productivity in Section 4.2. The nominal amount of time for the human-robot system to perform a specific task is labeled t_{task} in Figure 4-1a. The nominal amount of time required for the human agents to fulfill their part of the specific task is labeled $t_{human.finish}$. The time remaining once the human agents fulfill their part of the specific task is the time available for the humans to do other work, either within or outside the hostile operating environment.

In this chapter, robotic agents that do not require interventions are referred to as "reliable" while agents that do require interventions are referred to as "unreliable." These qualifiers, reliable and unreliable, are obviously not used in their traditional sense, but they take on an expanded meaning in which the underlying concept of failure (or time to failure) is replaced by the notion of intervention (or time to intervention). In the context of human-robotic systems, an *intervention* is not only driven by component failures, as discussed in Section 4.2.

The case where robotic agents are "unreliable" and require unplanned interventions is depicted in Figure 4-1b,c. Interventions while human agents are still fulfilling their part of the task increase $t_{human.finish}$ and t_{task}, as shown in Figure 4-1b. Unplanned interventions after $t_{human.finish}$ lead to a situation in which human agents may be required to remain in the hostile operational environment and attend to these

unplanned interventions. If human agents do not remain in the operational environment and a time penalty is incurred for repeatedly returning to the operational environment, the time required to respond and attend to interventions increases. This situation is depicted in Figure 4-1c. Each of these situations would significantly increase exposure and decrease the time available for human agents to do other work.

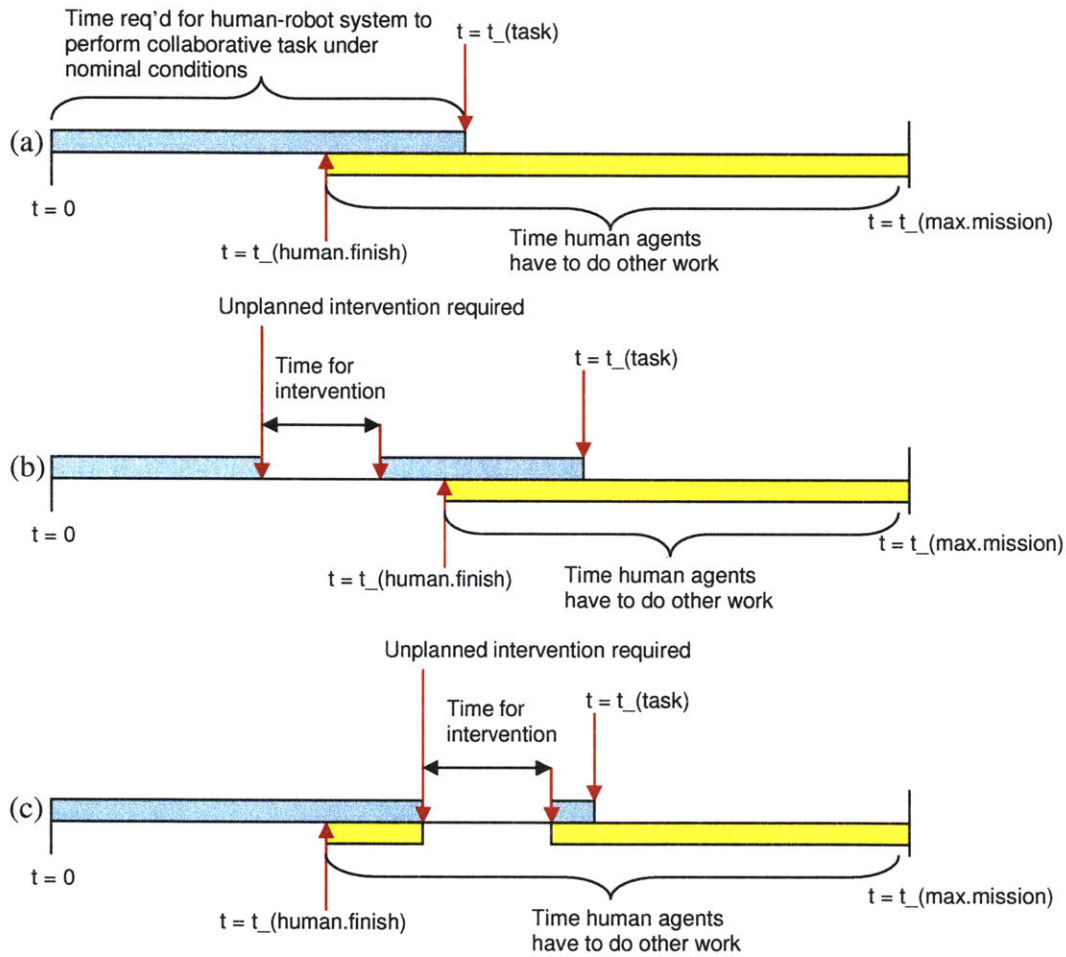

Figure 4-1: Effect of interventions on mission-level timeline

However, imagine that the human-robot system was "reliable" - in the sense that the robots did not often run into problems requiring intervention. In this case, the human agents would have a choice: once they finish their primary task, they could remain in the operational environment and begin working on other tasks. This would increase human productivity. Or, once the human agents finish their primary task,

they could return to the safe environment secure in the knowledge that the robots will continue to work without requiring interventions. This would minimize the exposure of the human agents. In other words, a "reliable" human-robot system provides the option of maximizing the effectiveness of the human-robot system by either maximizing human productivity or minimizing exposure. In the next section, these relationships are quantified using the metrics and definitions presented in Section 4.2.

4.4 Analytic formulation of the effect of interventions

In the following analyses, the effect of interventions on the objective of maximizing human productivity is explored by expressing the time available for humans to do other work as a function of MTBI and MTTI. In addition, the effect of interventions on the second objective discussed in Section 4.2, namely minimizing human exposure to the hostile environment, is explored by expressing the probability of intervention after the $t_{human.finish}$ as a function of MTBI and MTTI.

4.4.1 The effect of MTBI and MTTI on time for other work

Under nominal conditions (no interventions required), humans will complete their task in $t_{human.finish}$, as shown in Figure 4-1a. The remaining time, assuming the window of operation in the hostile environment $t_{max.mission}$ and nominal conditions, is given by,

$$t_{other} = t_{max.mission} - t_{human.finish} \tag{4.3}$$

As the robotic agents start requiring interventions, the time available for humans to do other work decreases, and thus limits the productivity of the human agents. This section explores how the time available for other work varies as a function of the mean time between intervention (MTBI) required by the robotic agents, and with the expected duration of interventions (MTTI).

A general expression of the time available for humans to do other work is presented in Eq. 4.4. This expression is normalized by the nominal amount of time (without interventions) for the human-robot system to perform a specific task. Time for other work is given by

$$\frac{t_{other}}{t_{task}} = 1 - \frac{t_{human.finish}}{t_{task}} - \frac{t_{human.finish}}{t_{task}MTBI}MTTI - \frac{t_{task} - t_{human.finish}}{t_{task}MTBI}MTTI \quad (4.4)$$

where $t_{human.finish}$ is the nominal amount of time (with no interventions) required for the human agents to fulfill their part of the specific task, and t_{task} is the nominal amount of time for the human-robot system to perform a specific task. The first two terms in Eq. 4.4 represent the time for other work with no interventions. The third term accounts for the total time spent engaging in interventions before $t_{human.finish}$, as depicted in Figure 4-1b. The last term accounts for the total time spent engaging in interventions after $t_{human.finish}$, as presented in Figure 4-1c.[3]

Figure 4-2 shows the amount of time human agents have available for other work as a function of MTBI and MTTI.

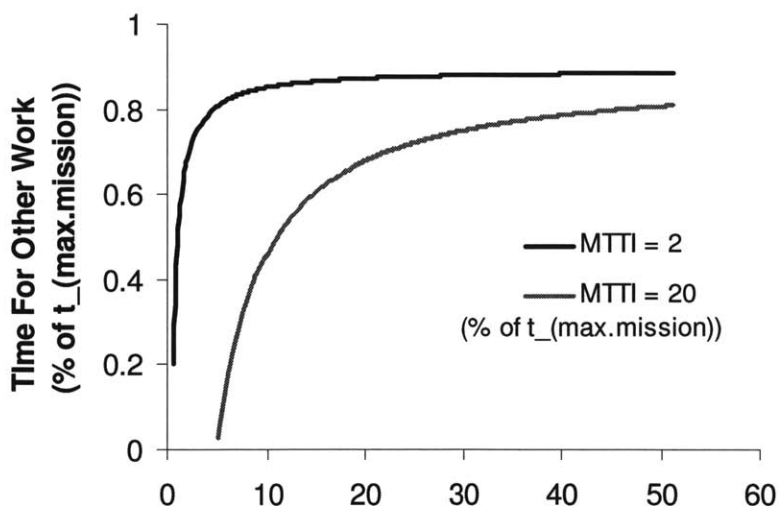

Figure 4-2: Time for other work as a function of MTBI and MTTI

[3]Eq. 4.4 also assumes that the Mean Time Between Intervention (MTBI) is measured from the end of one intervention to the start of the next intervention, and that two interventions cannot occur during the same time.

The figure shows that as MTBI increases, the Time For Other Work initially increases sharply and then plateaus. Increasing MTTI by an order of magnitude decreases the Time For Other Work and softens the transition between the initial increase and plateau as a function of MTBI.

This analysis assumed that MTTI and MTBI are constant throughout the mission duration, $t_{task} = 20\%$ of $t_{max.mission}$, and $t_{human.finish} = 10\%$ of $t_{max.mission}$. The curves represent the relationships between MTBI (ranging from 0 to 60%) and the specific MTTI (2% or 20%) for which the Time For Other Work is positive. The relationship between MTBI and MTTI which yield a positive Time For Other Work positive is given by

$$\frac{MTTI}{MTBI} \leq \frac{t_{max.mission} - t_{human.finish}}{t_{task}} \tag{4.5}$$

4.4.2 The effect of MTBI and MTTI on the probability of intervention

Once the human agents finish their part of the task, they may choose to exit the hostile environment as soon as possible to minimize exposure. Under nominal conditions, humans complete their part of the task in $t_{human.finish}$, and the robotic agents finish the task without requiring interventions as shown in Fig. 1a. In this case, the human agents may leave the hostile environment directly after $t_{human.finish}$. However, if the robotic agents are likely to require interventions after $t_{human.finish}$, the human agents may instead choose to remain in the hostile environment for a certain amount of time such that the probability of intervention past this point is within a specified threshold. This section explores the probability of intervention after $t_{human.finish}$ as a function of the mean time between intervention (MTBI) required by the robotic agents, and the expected duration of interventions (MTTI).

The probability of intervention after human agents finish their part of the task is described using a Poisson distribution to model the occurrence of initiating events. In this case, the initiating event is an intervention. The probability that at least one

intervention will be required between when the human agents finish their part of the task, and when the task is complete is given by

$$F(t) = 1 - exp\left[-\int_{t_{human.tot}}^{t_{task.tot}} h(s)ds\right] \tag{4.6}$$

where $t_{human.tot}$ is the time required (including interventions) for the human agents to perform their part of the task, and $t_{task.tot}$ is the total time (including interventions) to perform the task. Also, $h(s) = \frac{1}{MTBI}$ is constant, and MTTI is constant.

The resulting expression for the probability that at least one intervention will be required after $t_{human.finish}$ as a function of MTBI, MTTI, t_{task}, and $t_{human.finish}$ is given by,

$$F(t) = 1 - exp\left[-\frac{t_{task}}{MTBI} + \frac{t_{human.finish}}{MTBI} - \frac{t_{task} - t_{human.finish}}{MTBI^2}MTTI\right] \tag{4.7}$$

where $t_{human.finish}$ is the nominal amount of time (with no interventions) required for the human agents to fulfill their part of the specific task, and t_{task} is the nominal amount of time for the human-robot system to perform a specific task.

Figure 4-3 shows the probability that an unplanned intervention is required after the human agents have finished their part of the task as a function of MTBI and MTTI.

The figure shows that as MTBI increases, the probability of intervention decreases. Increasing MTTI by an order of magnitude shifts the curve and increases the probability of intervention for a given MTBI.

This analysis assumed that both MTBI and MTTI are expressed as a percentage of the total mission time, t_{task} = 20% of $t_{max.mission}$, and $t_{human.finish}$ = 10% of $t_{max.mission}$.

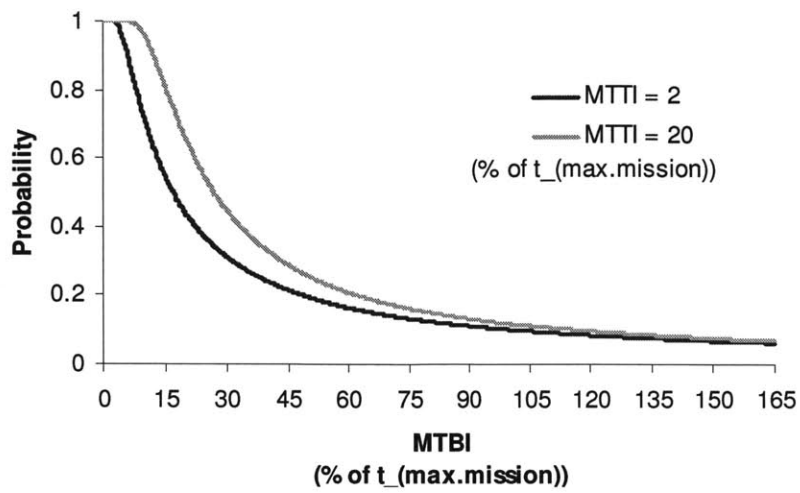

Figure 4-3: Probability of intervention as a function of MTBI and MTTI

4.5 Discussion

Preliminary analysis of the trends presented in the Section 4.4 yields interesting insights for design and evaluation of human-robot systems. Figure 4-2 indicates that the Time Available For Other Work is sensitive to both MTBI and MTTI. The sensitivities of Time Available For Other Work to changes in MTBI and MTTI are shown in Figure 4-4 and 4-5, and are respectively described by Eq. 4.8 and 4.9:

$$s_{MTTI} = \left| \frac{\partial t_{other}}{\partial MTTI} \right| = \frac{t_{task}}{MTBI} \tag{4.8}$$

and,

$$s_{MTBI} = \left| \frac{\partial t_{other}}{\partial MTBI} \right| = \frac{(MTTI)t_{task}}{MTBI^2} \tag{4.9}$$

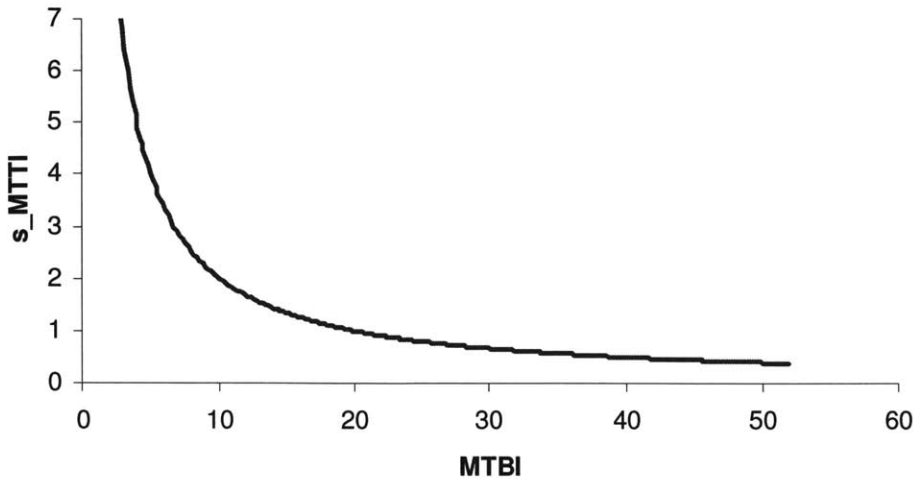

Figure 4-4: Sensitivity of time available for other work to MTTI

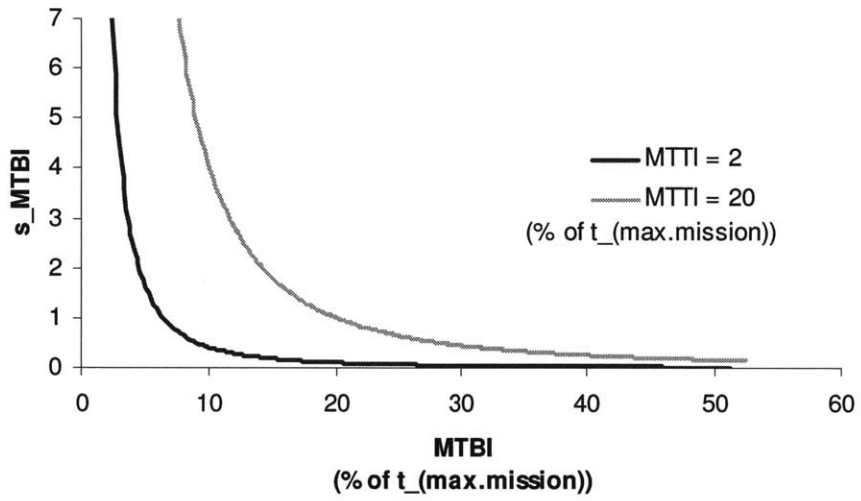

Figure 4-5: Sensitivity of time available for other work to MTBI

In Figure 4-4 and 4-5, t_{task}, MTBI, and MTTI are expressed as a percentage of the maximum mission time, and $t_{task} = 20\%$ of the maximum mission time.

Increases in MTTI reduce the time available for human agents to do other work. However this analysis shows that the sensitivity of Time Available For Other Work to MTTI is not a function of MTTI, and the sensitivity to MTTI quickly decreases as MTBI is increased. In other words, as the frequency of interventions decreases, the objective of maximizing human productivity becomes less sensitive to the duration of interventions. Also, increases in MTBI result in increased time available for other

work, and the sensitivity to MTBI increases as MTTI increases. In other words, increases in the duration of interventions results in greater sensitivity to the frequency of interventions. This suggests that a designer may be able to compensate for large or uncertain MTTI and achieve increases in time available to do other work with modest increases in MTBI.

Interestingly, Figure 4-3 indicates that the probability of robotic agents requiring an intervention after the astronauts have finished their part of the task is primarily a function of MTBI. The sensitivity of Probability of Intervention to changes in MTBI and MTTI are shown in Figure 4-6 and 4-7, and are respectively described by

$$s_{F-MTTI} = \left|\frac{\partial F(t)}{\partial MTBI}\right| = \frac{t_{task} - t_{human.finish}}{MTBI^2} exp\left[-\int_{t_{human.tot}}^{t_{task.tot}} h(s)ds\right] \tag{4.10}$$

and,

$$s_{F-MTBI} = \left|\frac{\partial F(t)}{\partial MTBI}\right| \tag{4.11}$$

$$= \left(\frac{t_{task}}{MTBI^2} - \frac{t_{human.finish}}{MTBI^2} + 2\frac{t_{task}-t_{human.finish}}{MTBI^3} MTTI\right) exp\left[-\int_{t_{human.tot}}^{t_{task.tot}} h(s)ds\right]$$

where t_{task} is the nominal amount of time (with no interventions) for the human-robot system to perform the task, $t_{human.finish}$ is the nominal amount of time (with no interventions) required for the human agents to fulfill their part of the specific task, $t_{human.tot}$ is the time required (including interventions) for the human agents to perform their part of the task, $t_{task.tot}$ is the total time (including interventions) to perform the task, and $h(s) = \frac{1}{MTBI}$.

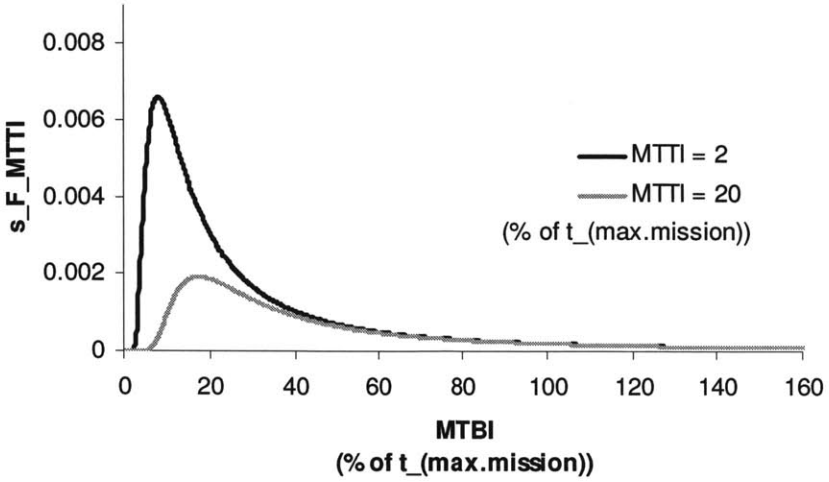

Figure 4-6: Sensitivity of probability of intervention to MTTI

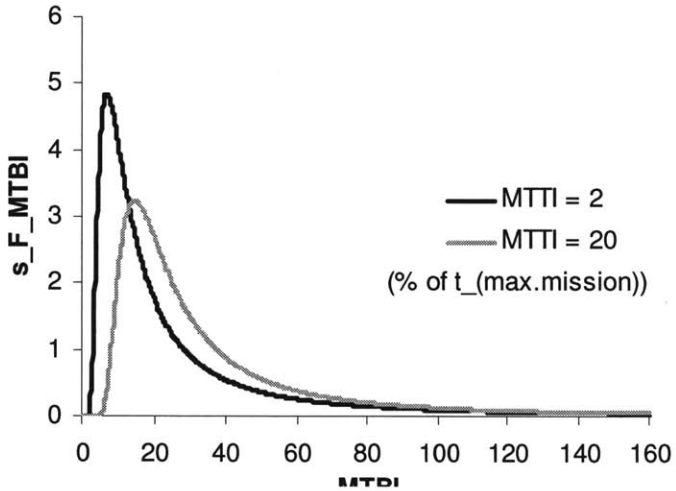

Figure 4-7: Sensitivity of probability of intervention to MTBI

In Figure 4-6 and 4-7, t_{task}, $t_{human.finish}$, MTBI, and MTTI are expressed as a percentage of the maximum mission time, $t_{task} = 20\%$, and $t_{human.finish} = 10\%$ of the maximum mission time.

This sensitivity analysis shows that the Probability of Intervention is nearly three orders of magnitude more sensitive to MTBI than MTTI. This suggests that MTBI is the primary driver, and unknown or uncertain MTTI may not significantly impact the design of a system to minimize the Probability of Intervention.

The meanings of MTBI and MTTI in the context of this analysis have implications for human-robot system design and evaluation. As discussed previously, MTBI is a function of:

1. The environment (and uncertainty in the environment) that the system or robotic agent is operating in.

2. The autonomy of the system or robotic agent, in this case defined as the ability to accommodate variations in the environment in pursuit of its goals citekn:revthirtysix (in this case, without human intervention).

3. The inherent component or system reliabilities

MTTI is a function of many variables, including (but not limited to):

- the nature of the failure or problem requiring an intervention
- the design parameters describing cognitive abilities and interaction among agents including:
  - the amount of information agents are able to gather about the nature of the failure prior to and during the intervention
  - the amount and type of information that can be transferred between a robotic and human agent.
- the physical distance between a robotic and human agent
- the lag in communications between a robotic agent and human agent
- the available resources and tools

This preliminary analysis suggests that a designer may be able to greatly impact both measures of the effectiveness of a human-robot system by increasing the MTBI, despite the likely variability and unpredictability of MTTI. In particular, the objective of maximizing human productivity becomes less sensitive to MTTI as MTBI increases. In addition, the objective of minimizing exposure is significantly more sensitive to

changes in MTBI than MTTI. These are encouraging results since MTBI is primarily a function of parameters that designers may influence, such as agent autonomy and component and system reliabilities.

These trends also have interesting implications for experiments aimed at evaluating the effectiveness of human-robot systems. Accurately characterizing MTTI through experimentation may not be necessary to formulate reasonable evaluations of the effectiveness of a human-robot system. This is fortunate since MTTI is dependent on a host of different factors and is likely to be difficult to accurately quantify. Another approach would be to characterize MTBI through experimentation with the factors that result in interventions, and conduct a sensitivity analysis to various MTTI.

Chapter 5

Case Study of the Peer-to-Peer Human-Robot Interaction Project

5.1 Introduction

In this chapter, I present a case-study of the Peer-to-Peer System, a multi-agent human-robot system developed at NASA. The purpose of this case-study is to provide a real-world example that illustrates the theoretical analysis presented in the previous chapter. During Fall 2005, I participated both in the planning of the data collection methodology at NIST, and the execution of the experiments with this system at the NASA Ames Research Center. I use the data collected during these experiments to evaluate the system using the methodology presented in the previous chapter. The objective of this analysis is to compare the Peer-to-Peer System's MTBI and MTTI (exhibited during the experiments) to the theoretical values of MTBI and MTTI necessary for the human-robot system to be more effective than a human-only system performing the experiment tasks.

In Section 5.2, I describe the Peer-to-Peer System and in Section 5.3, I give a formal description of the tasks performed during the experiment trials. In Section 5.4, I describe the data collected which is relevant to this thesis. In Section 5.5, I present the analysis of the data and analytical formulation for comparing the effectiveness of the system to a hypothetical human-only system. I also discuss limitations of this

analysis. In Section 5.6, I conclude and discuss how experimentation and the analysis can be expanded to provide a more accurate characterization of the Peer-to-Peer System's effectiveness (as defined in the previous chapter).

5.2 Description of system

The Peer-to-Peer Human-Robot Interaction (P2P-HRI) Project was developed at the NASA Ames Research Center in 2005 to support NASA's Vision for Space for a "sustained and affordable human and robotic program to explore the solar system and beyond. [50]" The purpose of Peer-to-Peer Human-Robot Interaction (P2P-HRI) project is to develop human-robot interaction techniques to allow humans and robots to work effectively together and compensate for limitations of robot autonomy through interaction. In particular the project has three objectives: (1) to develop natural language mechanisms for human-robot interaction, (2) to reduce workload for controlling robotic agents, and (3) to maximize the work that humans and robots accomplish together [32].

Experiments were conducted in November 2005 at NASA Ames with a multi-agent system to assess the project's progress with these objectives. The system consisted of five physical agents: two "EVA astronauts" (i.e. human agents in the operational environment), one "IVA astronaut" (i.e. human agent inside a habitat mockup), and two robotic agents. One of the robotic agents was Robonaut, a "torso-up" anthropomorphic humanoid robot developed at the Johnson Space Center [5]. Robonaut's fine motion and force-torque control allows it to perform many dexterous tasks often performed by humans. For these experiments Robonaut was mounted on a mobile platform, allowing it to autonomously traverse through the operational environment and perform pre-programmed actions. The other robotic agent was K-10, a mobile rover-like robot developed at NASA Ames with the capability to traverse and perform actions during the experiment trials autonomously or through tele-operation by EVA astronauts (with voice commands) or by the IVA astronaut (through a graphical interface).

A number of software agents made up the "Human-Robot Interaction Operating System" (HRI/OS), which enabled coordination and interaction among the physical agents. The software agents include: Task Manager, Context Manager, Resource Manager, Interaction Manager, and Spatial Reasoning Agent. The Task Manager coordinated actions by assigning high-level tasks to each of the agents. Each agent was then responsible for determining and executing the low-level actions necessary to complete the high-level task. Once the agent completed the assigned task, it reported the status back to the Context Manager. The Context Manager was responsible for maintaining a record of status, execution, and dialogue which other agents could query. The Resource Manager coordinated availability and requests for resources (agents and services) in the system. When an agent requested a resource, the Resource Manager generated a prioritized list of agents that should be consulted. The Interaction Manager then facilitated communication between agents through graphical interfaces and speech. Finally, the Spatial Reasoning Agent provided K-10 the ability to interpret voice commands (during tele-operation) using different frames of reference [33].

5.3 Experiment tasks

The November 2005 experiments included a number of trials in which the Peer-to-Peer System carried out a simulated construction project. During the construction activities, the EVA astronauts placed panels on a truss structure, Robonaut simulated welding the seams between panels, and K-10 simulated inspecting the integrity of each weld.

The formal description of the task specification that I have developed is presented in Tables 5.1 and 5.2. The start condition for the experiment trials included four of the six panels already mounted on the truss (with the other two panels located in the panel depot), and Seam 1-2 already welded. The *Enabling Precondition* for the astronauts to begin transporting the remaining panels was that Panels 5 and 6 were ready in the depot. Robotic agents conducted their functions (welding and inspecting)

with knowledge of an absolute frame of reference using the Visualeyez System, a 3-D motion capture system [76]. The Visualeyez system provided the location of each agent in the operational environment, and the placement of the truss structure which held the panels was fixed within the absolute frame of reference. Robonaut and K-10 were not able to "sense" the panels or identify variations in panel and seam placement from the expected location. As a result, proper operation of the Visualeyez system was a necessary *Permanent Precondition* for all functions to avoid agent collisions and to achieve well-placed welds and proper inspections. Other *Permanent Preconditions* included the necessity for the frames holding the panels to be precisely assembled and placed within the absolute frame of reference, and then for the panels to be precisely placed on the frames. The sources of error associated with uncertainty in each of these permanent preconditions had to sum to less than 7-8 inches for Robonaut to successfully weld each seam. *Ordering Constraints* are also presented in the formal task specification.

While this task specification presents the baseline scenario for the experiment trials, two scripted "unplanned interventions" were carried out during each trial. I refer to these two scripted "unplanned interventions" as "planned interventions" to distinguish them from the truly unplanned interventions discussed in Sections 5.4 and 5.5. One planned intervention occurred while the astronauts were mounting Panel 5. During this first intervention, the EVA astronauts realized they needed a light to better check the placement of Panel 5. One of the EVA astronauts then utilized the Resource Manager and Interaction Manager to initiate voice-command tele-operation of K-10. The purpose of this intervention was to exercise the Spatial Reasoning Agent and command K-10 to shine its light on Panel 5. The second planned intervention occurred while Robonaut was welding either Seam 3-4 or 4-5. For this intervention, Robonaut would stop in the middle of its weld and request that an EVA astronaut check the placement of a panel.

Final Outcome of Portfolio of Tasks: All six panels placed on truss; Seams 1-2, 2-3, 3-4, and 4-5 welded; and Seams 1-2, 2-3, and 3-4 inspected.

SUBTASKS		ENVIRONMENT				ORDERING CONSTRAINTS			
Function	**Uncertainty**	**Permanent Preconditions**		**Enabling Preconditions**		**Pre**	**Post**	**Concurrent**	
		State	**Uncertainty**	**State**	**Uncertainty**				
Transport Panel 5 to Position 5	None	3-D location of robotic and/or human agents known through Visualeyez system	Visualeyez operational during trials?	Panels are in the Panel Depot	None	None	Place Panel 5 on Frame 5	Weld Seam 2-3, 3-4	Inspect Seam 1-2, 2-3, 3-4
Transport Panel 6 to Position 6							Place Panel 6 on Frame 6	Weld Seam 2-3, 3-4, 4-5	Inspect Seam 1-2, 2-3, 3-4
Place Panel 5 on Frame 5	None	3-D location of robotic and/or human agents known through Visualeyez system	Visualeyez operational during trials?	None	None	Transport Panel 5 to Position 5	Transport Panel 6 to Position 6	Weld Seam 2-3, 3-4, 4-5	Inspect Seam 1-2, 2-3, 3-4
		Frames are ideally assembled and placed	Sources of error in frame assembly and placement add up to less than 7-8 inches?						
Place Panel 6 on Frame 6		Position and orientation of ideal frames known through Visualeyez	Visualeyez operational during trials?			Transport Panel 6 to Position 6	None	Weld Seam 2-3, 3-4, 4-5	Inspect Seam 1-2, 2-3, 3-4
		Lighting	Lighting is adequate?						
Weld Seam 2-3	None	3-D location of robotic and/or human agents known through Visualeyez system	Visualeyez operational during trials?	None	None	Transport Panel 2,3 to Position 2,3; Place Panel 2,3 on Frame 2,3	Weld Seam 3-4	Transport Panel 5,6 to Position 5,6; Place Panel 5,6 on Frame 5,6	Inspect Seam 1-2

Table 5.1: P2P-HRI system experiment task specification

		Frames are ideally assembled and placed	Sources of error in frame assembly and placement add up to less than 7-8 inches?							
		Panel ideally placed and secured on frame	Variation in panel placement?							
Weld Seam 3-4		Position and orientation of ideal frames known through Visualeyez	Visualeyez operational during trials?			Transport Panel 3,4 to Position 3,4	Place Panel 3,4 on Frame 3,4	Weld Seam 4-5	Transport Panel 5,6 to Position 5,6	Place Panel 5,6 on Frame 5,6	Inspect Seam 1-2, 2-3
Weld Seam 4-5						Transport Panel 4,5 to Position 4,5	Place Panel 4,5 on Frame 4,5	None	Transport Panel 6 to Position 6	Place Panel 6 on Frame 6	Inspect Seam 1-2, 2-3, 3-4
Inspect Seam 1-2	None	3-D location of robotic and/or human agents known through Visualeyez system	Visualeyez operational during trials?	None	None	Weld Seam 1-2		None	Transport Panel 5,6 to Position 5,6	Place Panel 5,6 on Frame 5,6	Weld Seam 2-3, 3-4, 4-5
Inspect Seam 2-3						Weld Seam 2-3			Transport Panel 5,6 to Position 5,6	Place Panel 5,6 on Frame 5,6	Weld Seam 3-4, 4-5
Inspect Seam 3-4		Frames are ideally assembled and placed	Sources of error in frame assembly and placement add up to less than 7-8 inches?			Weld Seam 3-4			Transport Panel 5,6 to Position 5,6	Place Panel 5,6 on Frame 5,6	Weld Seam 4-5
Inspect Seam 4-5		Panel ideally placed and secured on frame	Variation in panel placement?			None			None		

Table 5.2: P2P-HRI system experiment task specification, cont.

5.4 Data collection methods

Data was collected in the form of video, audio, and real-time manual data logging for seven experiment trials. Four cameras were positioned to capture video and audio during the trials. Three of these cameras were placed in the operational area to capture the movements and audio of the two EVA astronauts, Robonaut, and K-10. The fourth camera was placed inside the habitat to capture the IVA astronaut's voice, actions, and the graphical interfaces used to monitor the system. The camera inside the habitat also captured the EVA astronaut's audio feed. Figure 5-1 shows the camera placement in the operational environment and habitat.

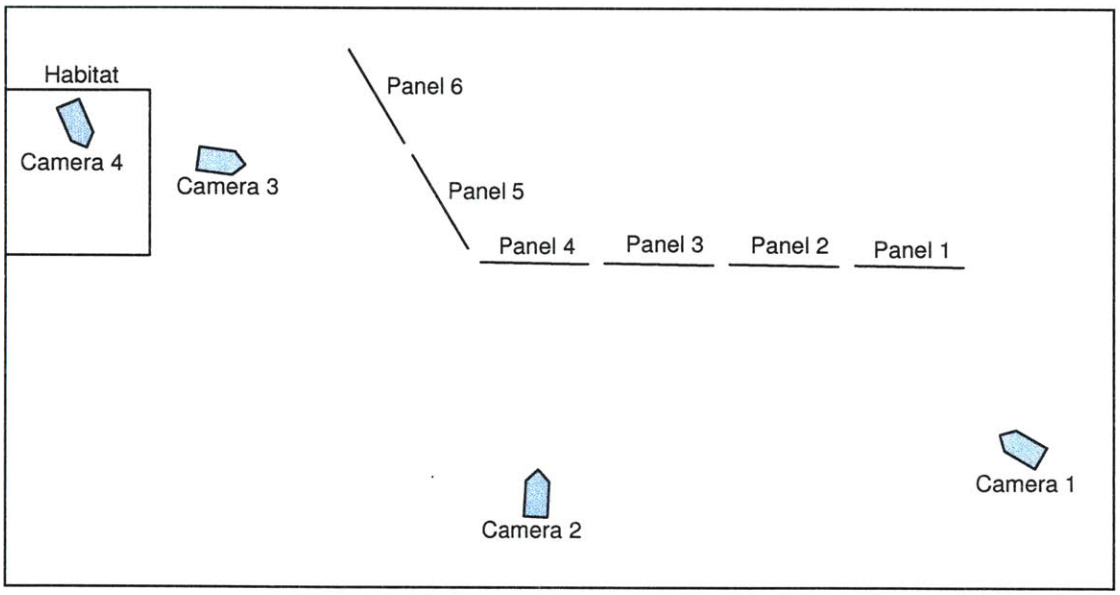

Figure 5-1: Overhead view of camera placement

Real-time manual data logging was also conducted to complement the video and audio data. Noldus information technology software specifically designed to collect and analyze behavioral data was used to log the timing of each agent's actions [54]. Three people manually logged data during each trial, myself and the two researchers from NIST. One person recorded the actions of Robonaut, the second person recorded the actions of K10, and the third person recorded the actions of both EVA astronauts. The actions of the IVA astronaut were not logged using the Noldus software. Table

5.3 lists the actions recorded for each agent.

EVA Astronaut Actions	K-10 Actions	Robonaut Actions
Start Get Panel	Start Move	Start Move
End Get Panel	End Move	End Move
Start Place Panel	Start Inspection	Start Weld
End Place Panel	End Inspection	End Weld
Start Request Help	Start Provide Help	Start Request Help
End Request Help	End Provide Help	End Request Help
Start Provide Help	Interesting Event Marker	Interesting Event Marker
End Provide Help		
Interesting Event Marker		

Table 5.3: Experiment actions

At the beginning of each trial, the Noldus software was synchronized with the cameras. During the trials, the start time and stop time was recorded for each action, and the Noldus software then computed the duration of each recorded action. The purpose of these measurements was to provide approximate timing for actions as well as provide pointers to the camera data for interesting events.

5.5 Data analysis

The objective of this analysis is to compare the Peer-to-Peer System's MTBI and MTTI (exhibited during the experiments) to the theoretical values of MTBI and MTTI necessary for the human-robot system to be more effective than a human-only system performing the experiment tasks.

Video and Noldus data were collected for seven trials. Each of these trials was analyzed to identify unplanned interventions, and to estimate the MTBI, MTTI, t_{task}, and $t_{human.finish}$ (Section 5.5.1). A hypothetical human-only workflow to carry out the experiment task was then derived (Section 5.5.2). Lastly, the experiment values

of MTBI and MTTI were compared to the theoretical values necessary for the human-robot system to provide a higher Time For Other Work than the human-only baseline, and achieve a Probability of Intervention of less than 25% (Section 5.5.3).

5.5.1 Unplanned interventions and experiment values for MTBI, MTTI

Table 5.4 lists each of the unplanned interventions, and descriptive information including: the experiment trial the intervention occurred in, duration, circumstances, the agent requesting help, and the agents providing help.

Based on these thirteen unplanned interventions, experiment MTBI and MTTI were derived from the video data and are shown in Table 5.5. The mean total time to complete the task, t_{task}, is 710 seconds, and the mean time required for the humans to complete their part of the task, $t_{human.finish}$, is 471 seconds. Although the experiment did not have maximum mission time, I assumed for the purposes of analysis that $t_{max.mission}$ is 3550 seconds, five times the mean duration of t_{task}.

5.5.2 Hypothetical human-only workflow

In the previous section I analyzed experiment data to determine the MTBI and MTTI of the P2P-HRI. In this section I develop a hypothetical human-only workflow to provide a basis of comparison for measuring the effectiveness of the human-robot system. A human-only workflow would not involve robotic agents, and therefore would not have any unplanned interventions in which a robotic agent requests help from a human agent. To provide a conservative basis for comparison, I assumed that the time required for humans to perform the primitive tasks (such as weld, inspect, move between panels) is equal to the time the robotic agents take to perform the same primitive tasks. I suggest that this is a fair basis for comparison, since the experiment trials did not involve real welding or inspections, making it difficult to compare the inherent capabilities of the humans and robots for performing these sorts of primitive tasks.

Incident	Trial #	Duration (sec)	Circumstances	Request From	Help Given By
K10 not responding	1	17	EVA1 request for tele-operation (during EVA1's request for light)	K10	EVA1, IVA
K10 encounters obstacle	1	11	K10 encounters obstacle (Robonaut) while trying to return to its task after providing flashlight	K10	EVA1, IVA
K10 encounters obstacle	2	67	K10 gets too close to Robonaut while traversing to help EVA1&2 with light	K10	EVA1, IVA
K10 encounters obstacle	2	20	EVA1&2 are too close to K10 when they end tele-operation and sent K10 back to its task	K10	EVA1, IVA
K10 encounters obstacle	2	68	K10 encounters obstacle (Robonaut) while trying to return to its task after providing flashlight	K10	IVA
Task Manager fails	2	130	Task Manager fails to give Robonaut and EVA1&2 next task	K10, EVA1&2	IVA
K10 encounters obstacle	3	141	K10 is too close to Robonaut. Concurrently, there is a problem with K10's spatial reasoning	K10	EVA1, IVA
K10 encounters obstacle	4	37	K10 encounter an obstacle (Robonaut) while traversing to help EVA1&2 with light	K10	EVA2, IVA
K10 not responding	4	168	K10 stops along its traverse back to its task after providing light	K10	IVA
K10 encounters obstacle	5	22	K10 encounters an obstacle (Robonaut) while traversing to help EVA1&2 with light	K10	EVA1
K10 encounters obstacle	6	29	K10 encounters an obstacle (Robonaut) while traversing to help EVA1&2 with light	K10	EVA2, IVA
K10 encounters obstacle	6	38	K10 encounters an obstacle (Robonaut) while returning after helping EVA1&2 with light	K10	IVA
K10 not responding	7	61	Lost track of K10 while it was traversing to help EVA1&2 with light	K10	EVA1, IVA

Table 5.4: P2P-HRI experiment unplanned interventions

Experiment Parameters	**Mean** (seconds)	**Standard Deviation** (seconds)
MTBI	115	74
MTTI	67	53

Table 5.5: Experiment MTBI and MTTI

In developing the human-only baseline, I assumed that two humans are required to mount a panel, and one human is necessary to either weld or inspect a seam. As in the experiment scenario, I assumed that Panels 1 through 4 have been mounted, Seam 1-2 has been welded, and Seam 5-6 does not need to be welded or inspected. The planned interventions built into the experiment trials are also incorporated into the hypothetical human-only baseline. For instance, mounting panel five includes procuring a light to check the alignment of the panel. The hypothetical time required for the human-only team to find a light and inspect the alignment is assumed to be equal to the time for the planned intervention in which K10 provides a light to the EVA astronauts. Similarly, welding Seam 4-5 includes the time for the planned intervention in which Robonaut requests assistance from EVA1 and EVA2.

Table 5.6 lists the timing data for the primitive tasks which make up the human-only workflow. Five instances (labeled 1-5 in the table), each chosen randomly from different trials, were recorded. The timing data for these primitive tasks was used to formulate a specific hypothetical human-only workflow shown in Table 5.7. It is important to note that this may not be the optimum workflow, but is one reasonable workflow which is used as a baseline for this analysis. The total task time and maximum time for other work are calculated assuming the mean time (labeled as Average), one standard deviation below the mean time (labeled as Low), and one standard deviation above the mean time (labeled as High) for each primitive task.

	1 (sec)	**2** (sec)	**3** (sec)	**4** (sec)	**5** (sec)	**Average** (sec)	**Standard Deviation** (sec)
Mount Panel 5 including light intervention	505	339	225	239	287	319	113
Mount Panel 6	45	47	46	40	42	44	3
Weld Seam 4-5 including planned intervention (includes traveling TO)	115	232	205	113	215	176	57
Weld Seam (w/o intervention) (includes traveling TO)	123	111	108	93	106	108	11
Inspect Seam (includes traveling TO)	30	32	67	30	27	37	17

Table 5.6: Timing data for primitive tasks

Primitive Tasks – EVA1	**Average** (sec)	**Low** (sec)	**High** (sec)	**Primitive Tasks – EVA2**	**Average** (sec)	**Low** (sec)	**High** (sec)
Mount Panel 6	44	41	47	Mount Panel 6	44	41	47
Mount Panel 5	319	206	432	Mount Panel 5	319	206	432
Weld Seam 4-5	176	119	233	Idle	176	119	233
Weld Seam 3-5	108	97	119	Inspect Seam 4-5	37 (Idle: 71)	20 (Idle: 88)	54 (Idle: 54)
Weld Seam 2-3	108	97	119	Inspect Seam 3-4	37 (Idle: 71)	20 (Idle: 88)	54 (Idle: 54)
Inspect Seam 1-2	37	20	54	Inspect Seam 2-3	37	20	54
Total Task Time	**792**	**580**	**1004**				
Max Time for Other Work	**2758**	**2970**	**2546**				

Table 5.7: Hypothetical human-only workflow

5.5.3 Analysis of P2P-HRI system effectiveness

The previous two sections provide the basis to analyze the effectiveness of the P2P-HRI system. In this section I analyze the effectiveness of the P2P-HRI System using two metrics: time for humans to do other work as compared to the human-only baseline, and the probability of unplanned intervention after the humans finish their part of the task. The analytic formulation for each of these metrics described in Chapter 4 is applied in this section.

Figure 5-2 compares the Time For Other Work as a function of the MTBI and MTTI exhibited during the experiment trials to the Time For Other Work of the hypothetical human-only baseline.

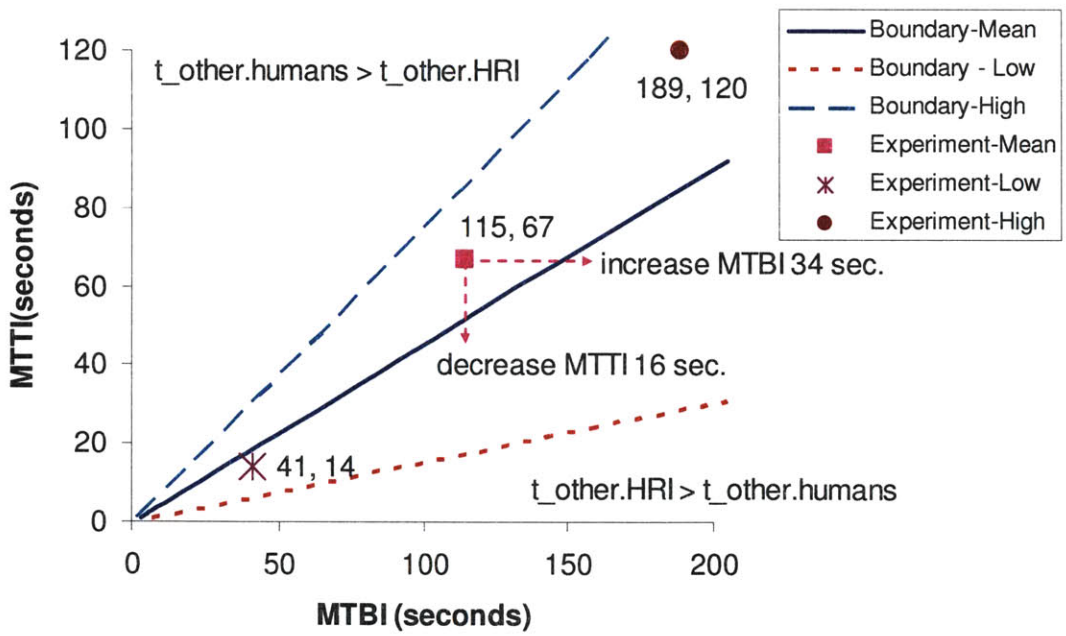

Figure 5-2: Time for other work of P2P-HRI System compared to human-only baseline

This analysis assumed t_{task} = 710 seconds, $t_{human.finish}$ = 471 seconds, and $t_{max.mission}$ = 3550 seconds. The Boundary-Mean line indicates the boundary over which the human-only baseline has a greater Time For Other Work than the P2P-HRI System. The Boundary-Mean line is calculated using the mean values for prim-

itive tasks to formulate the human-only baseline (calculated in Section 5.5.2). The Boundary-Low line is calculated using the low values for primitive tasks, and the Boundary-High line is calculated using the high values for primitive tasks. The Experiment-Mean data point indicates the mean values of MTBI and MTTI exhibited during the experiment trials. Due to the large standard deviation of MTBI and MTTI, I have also plotted Experiment-Low and Experiment-High data points. Experiment-Low indicates one standard deviation below the mean for both MTBI and MTTI, and Experiment-High indicates one standard deviation above the mean for both MTBI and MTTI.

The figure shows that the Experiment-Mean MTBI must be increased by 34 seconds or the Experiment-Mean MTTI must be decreased by 16 seconds to cross the Boundary-Mean line, making the Time For Other Work of the P2P-HRI System greater than for the human-only baseline. These are relatively modest improvements in MTBI and MTTI. Additionally, all three Experiment-Mean, Low, and High data points fall within the Boundary-High line. This evidence suggests that the P2P-HRI System and the human-only baseline have comparable effectiveness in terms of the metric Time For Other Work.

The second metric of effectiveness is the probability that an unplanned intervention will be required after the $t_{human.finish}$. Figure 5-3 compares the MTBI and MTTI exhibited in the experiment to a Probability of Intervention less than 25%.

This analysis assumed $t_{task} = 710$ seconds, and $t_{human.finish} = 471$ seconds. The Boundary line indicates the MTBI and MTTI which result in either a Probability Of Intervention greater than 25% or less than 25%. The Experiment-Mean data point indicates that mean values of MTBI and MTTI exhibited during the experiment trials. The Experiment-Mean MTBI and MTTI result in greater than 96% chance of an intervention after $t_{human.finish}$. As with the previous analysis, I have also plotted Experiment-Low and Experiment-High data points. Experiment-Low indicates one standard deviation below the mean for both MTBI and MTTI, and result in greater than 99% chance of an intervention after $t_{human.finish}$. Experiment-High indicates one standard deviation above the mean for both MTBI and MTTI, and result in greater

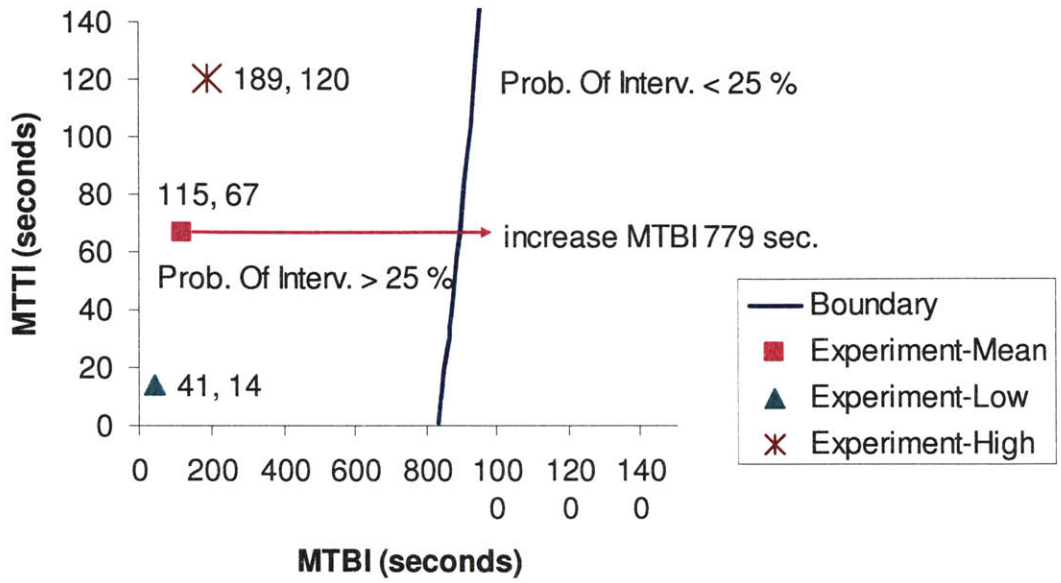

Figure 5-3: Probability of intervention in P2P-HRI System

than 87% chance of an intervention after $t_{human.finish}$.

These results indicate that although the P2P-HRI System is comparable to a human-only system in terms of Time For Other Work, the Probability of Intervention is significant. In fact, the Experiment-Mean MTBI must be increased by 779 seconds for the Probability of Intervention to drop below 25%. This suggests that unless significant improvements can be made in the MTBI, EVA astronauts should spent their Time For Other Work in the operational environment, rather than returning inside the habitat.

A potential shortcoming of this analysis is identified by comparing the analysis of Probability of Intervention to the actual occurrences in interventions after $t_{human.finish}$. In fact out of seven trials, an intervention after $t_{human.finish}$ only occurred once. This does not seem consistent with Probabilities of Intervention which range from 87-99%. The reason for this discrepancy is that interventions tended to occur during particular parts of the experiment trial involving human-robot interaction. Since there was no human-robot interaction after $t_{human.finish}$, the actual MTBI after $t_{human.finish}$ was much lower than the overall MTBI. Based on this observation,

the MTBI-MTTI analysis is likely to be most descriptive for homogeneous tasks in which the MTBI and MTTI remain constant over the task timeline. In absence of homogeneous tasks, the MTBI-MTTI analysis can be expanded to explicitly account for quantitatively different phases in the task and the duration of each phase.

5.6 Conclusions

In this chapter, I have presented a case study of the Peer-to-Peer Human-Robot Interaction System with the purpose of providing a real-world example to illustrate the analysis of HRI system effectiveness presented in Chapter 4. I have described the system, including agents and their roles during the November 2005 experiments. I have formally described the experiment tasks, including functions, permanent and enabling preconditions, and ordering constraints, as well as described the modes of data collection. In analyzing the data, I identified unplanned interventions and derived experimental values for MTBI and MTTI, formulated a hypothetical human-only baseline, and analyzed the effectiveness of the P2P-HRI System.

The effectiveness of the P2P-HRI System was analyzed using two metrics, Time For Other Work and Probability of Intervention, using the analytic formulation presented in Chapter 4. In analyzing the time humans have available for work in the P2P-HRI System as compared to a human-only baseline, I concluded that the Time For Other Work in the P2P-HRI System is comparable to the human-only baseline. Indeed, modest improvements in either MTBI or MTTI would result in an HRI System that is more effective than the human-only baseline, with respect to Time For Other Work. On the other hand, the analytic formulation of Probability of Intervention suggests that there is an 87-99% Probability of Intervention after $t_{human.finish}$. In this case, a significant increase in MTBI is necessary to achieve less than 25% Probability of Intervention. The difference in system effectiveness derived from the two measures suggests that both metrics must be considered to form an accurate picture of the capabilities and limitations of a human-robot system. Together, these measures of system effectiveness capture the coupled relationships between, reliabil-

ity, productivity, and risk to humans for human-robot systems, and provide high-level metrics to compare the performance of different systems.

Chapter 6

Conclusion

6.1 Summary of contributions

My overall research objective is to the lay the foundations of a unified framework for architecting human-robotic systems for optimal task performance given a set of metrics. In this thesis, I have addressed three issues to accomplish this objective. (1) What are the considerations for architecting human-robot teams? (2) How can a designer systematically and formally capture the characteristics that describe each human-robotic system to ensure that the evaluations of systems are comparable? (3) How can a designer analytically formulate common metrics to evaluate and maximize the effectiveness of a human-robot system?

In addressing the first question, I reviewed literature from different fields including HRI and human-computer interaction, and synthesized multiple considerations for architecting heterogeneous teams of humans and robots (Chapter 2). I organized these considerations into four main building blocks to provide the backbone for an analytical framework to compare the advantages and disadvantages of different human-robot systems. The building blocks included: 1) specifying tasks, 2) generating a human-robotic team, 3) allocating functions to agents in the system, and 4) evaluating the system against common task-based metrics.

I further developed the second building block, generating a human-robotic team, and addressed how to capture the characteristics that describe a human-robotic sys-

tem (Chapter 3). Methods for formally describing human-robot systems include: decomposing the system-independent task specification into system-specific functions with ordering and timing constraints (1); identifying the interactions necessary to fulfill system-specific functions as well as for specifying the characteristics of these interactions (2); specifying how system-specific components of form are utilized to fulfill system-specific functions (3); and identifying the design parameters relevant to analyzing the autonomy of the system (4). I then illustrated these methods by formally describing the architecture of the Nursebot System.

To address the last issue of formulating common metrics, I proposed an analytical formulation of common metrics to guide the design and evaluate the performance of human-robot systems (Chapter 4). I presented objectives for maximizing the effectiveness of a human-robot system which capture the coupled relationships among productivity, reliability, and risk to humans. I developed reliability parameters to characterize unplanned interventions between a human and robot, and then analytically investigated the effect of unplanned interventions on the effectiveness of human-robot systems using traditional reliability analysis. Finally, I applied this analysis to data collected during experiment trials with the NASA Peer-to-Peer Human-Robot Interaction System (Chapter 5), and compared the system performance to the hypothetical performance of a human-only team conducting the same tasks. Based on the results, I proposed recommendations for quantifying future improvements in effectiveness of the human-robot system.

6.2 Future work

There exist many directions for future research and potential applications of such a framework in space exploration and other fields. Further development of the building block 3) task allocation is one exciting area open for future work. In particular, the framework would benefit from research into a tractable means of applying AI planning and scheduling techniques to represent human-robotic systems in terms of complex mixtures of action choices, ordering constraints, and metric quantities to enumerate

and score different task allocation options.

A key to the success of the AI planning and scheduling approach to task allocation is to develop meaningful metric quantities to associate with action choices. For such a framework to fully characterize the effectiveness of a system, metric quantities need to be developed for productivity/effectiveness, reliability, risk to humans, resources, and flexibility/robustness. This issue was not addressed in this thesis; instead, global parameters such as Mean Time Between Interaction (MTBI) and Mean Time To Interact (MTTI) were defined to address issues of reliability, productivity, and risk to humans. Future research should work towards developing an experimental-based understanding of lower-level metrics (such as situational awareness in human-robot systems). These lower-level metrics may prove useful in providing analytical relationships between the description of system architecture and high-level metrics, and in developing meaningful metrics quantities to associate with specific action choices.

A unified framework to optimally leverage the capabilities of humans and robots will be an invaluable asset to decision-makers and designers of human-robot systems in a variety of different fields including space exploration, military operations, civilian search and rescue operations, healthcare settings, and other areas. Decision-makers may use this framework as a tool to analytically investigate the trade-offs among different human-robot systems and inform decisions on which system to invest in or implement. System designers may also use this framework to analyze the performance of a particular human-robot system with the purpose of identifying and quantifying necessary improvements to make their system a competitive option. Finally, the further work in task allocation will be applicable to real-time operations of human-robot systems as well as for design and evaluation of systems. An AI planning and scheduling task allocation method which incorporates metric quantities for evaluating high-level common metrics could be used to implement systems which are capable of choosing task allocations to maximize the specific metrics which are appropriate during the different phases of a mission timeline. This capability would be an invaluable asset for mission planning and execution in space and terrestrial applications.

Bibliography

[1] A. Acquisti, M. Sierhuis, W. Clancey, and J. Bradshaw. Agent based modeling of collaboration and work practices onboard the international space station. In *Proceedings of the Eleventh Conference on Computer-Generated Forces and Behavior Representation*, Orlando, FL, 2002.

[2] F. Aghazadeh, R. Chapleski, and R. Hirschfeld. A hazard analysis system for robotic work cells. *Human Factors and Ergonomics in Manufacturing*, 8(4):232–230, 1998.

[3] J. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[4] Mark Allender. Sts-49, the rescue of intelsat-vi. http://uu.cx/flight/49/. Accessed: 5/05.

[5] R. Ambrose and et al. H. Aldridge. Robonaut: Nasa's space humanoid. *IEEE Intelligent Systems Journal*, 2000.

[6] G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, N. Roy, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In *Workshop on Interactive Robots and Entertainment*, 2000.

[7] K. S. Barber and C. E. Martin. Agent autonomy: Specification, measurement, and dynamic adjustment. In *Proceedings of the Autonomy Control Software Workshop at Autonomous Agents (Agents'99)*, Seattle, WA, 1999.

[8] K.S. Barber, A. Goel, T.J. Graser, T.H. Liu, R.H. Macfadzean, C.E. Martin, and S. Ramaswamy. Sensible agents. In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, FL, 1997.

[9] J.E. Bardram. Plans as situated action: An activity theory approach to workflow systems. In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work ECSCW'97 Conference*, Dordrecht, Holland, 1997.

[10] S. Brainov and H. Hexmoor. Quantifying relative autonomy in multiagent interaction. In *Proceedings of the UJCAI'01 Workshop Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, Seattle, WA, 2001.

[11] J. Bresina, K. Golden, D.E. Smith, and R. Washington. Increased flexibility and robustness of mars rovers. In *Fifth Intl. Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, Netherlands, 1999.

[12] J. Burke, R. R. Murphy, E. Rogers, J. Scholtz, and V. Lumelsky. Final report for the darpa/nsf interdisciplinary study on human-robot interaction. *IEEE Systems, Man and Cybernetics Part C*, 34(2):103–112, 2004.

[13] R. Burridge, J. Graham, K. Shillcutt, R. Hirsh, and D. Kortenkamp. Experiments with an eva assistant robot. In *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS-03)*, Nara, Japan, 2003.

[14] J. Carleson and R. Murphy. Human-robot interaction in rescue robotics. *IEEE Trans. Systems, Man, and Cybernetics*, 34(2):138–153, 2004.

[15] J. Carlson and R.R. Murphy. Reliability analysis of mobile robots. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003.

[16] C. Carreras and I. D. Walker. Interval methods for fault-tree analysis in robotics. *IEEE Trans. Robot. Automat*, 50(1):3–11, 2001.

[17] R. Chatila, S. Lacroix, T. Simeon, and M. Herrb. Planetary exploration by a mobile robot: mission teleprogramming and autonomous navigation. *Autonomous Robots*, 17(4):315–337, 1998.

[18] E. Coste-Maniere, B. Espiau, and E. Rutten. A task-level robot programming language and its reactive execution. In *Proceedings of the 1992 IEEE International Conference on Robotics and Autonomy*, Nice, France, 1992.

[19] C. Culbert, J. Rochlis, F. Rehnmark, and D. Kortenkamp et.al. Activities of the nasa exploration team human-robotics working group. In *Space 2003*, Long Beach, CA, September 2003.

[20] Jr. D. R. Olsen and M. A. Goodrich. Metrics for evaluating human-robot interaction. In *Proceedings of PERMIS 2003*, Gaithersburg, MD, 2003.

[21] K. Dautenhahn, B. Ogden, and T. Quick. From embodied to socially embedded agents. *Journal of Cognitive Systems Research*, 3(3):397–428, 2002.

[22] B.S. Dhillon and A.R.M. Fashandi. Safety and reliability assessment techniques in robotics. *Robotica*, 15:701–708, 1997.

[23] B.S. Dhillon and N. Yang. Reliability analysis of a repairable robot system. *Journal of Quality in Maintenance Engineering*, 2(2):30–37, 1996.

[24] D. Dori. *Object-Process Methodology - A Holistic Systems Paradigm.* Springer, 2002.

[25] J. Drury, J. Scholtz, and H. Yanco. Awareness in human-robot interactions. In *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, pages 912–918, 2003.

[26] J. Drury, J. Scholtz, and H. Yanco. Applying cscw and hci techniques to human-robot interaction. In *Proc. CHI 2004 Workshop on Shaping Human-Robot Interaction*, Vienna, Austria, 2004.

[27] R. Effinger, A. Hofmann, and B. Williams. Progress towards task-level collaboration between astronauts and their robotic assistants. In *Proc. ISAIRAS 2005 Conference*, Munich, Germany, 2005.

[28]] K. Erol, J. Hendler, and D. Nau. Umcp: A sound and complete procedure for hierarchical task-network planning. In *Proc. 2nd Intl. Conf. on A.I. Planning Systems*, pages 249–254, 1994.

[29] K. Erol, J. Hendler, and D. Nau. Semantics for hierarchical task network planning. Technical report CSTR-3239, 1994. Dept. of Computer Science, U. Maryland.

[30] K. Ikuta et. al. Safety evaluation method of human-care robot control. In *Proc. Intl. Symp. Micromechatronics and Human Science*, pages 119–127, 2000.

[31] T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and A. Steinfeld. Common metrics for human-robot interaction. In *IEEE International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[32] T. Fong, I. Nourbaksh, C. Kunz, L. Fluckiger, J. Schreiner, R. Ambrose, R. Burridge, R. Simmons, L Hiatt, A. Schultz, J.G. Trafton, M. Bugajska, and J. Scholtz. The peer-to-peer human-robot interaction project. In *AIAA Space 2005*, 2005.

[33] T. Fong, J. Scholtz, J.A. Shah, L. Fluckiger, C. Kunz, D. Lees an J. Schreiner, M. Siegel, L. Hiatt, I. Nourbakhsh, R. Simmons, R. Ambrose, R. Burridge, B. Atonishek, M. Bugajska, A. Schultz, and J.G. Trafton. A preliminary study of peer-to-peer human-robot interaction. In *System, Man, and Cybernetics Conference 2006*, Taipei, Taiwan, 2006.

[34] T. Fong, C. Thorpe, and C. Baur. Robot as partner: Vehicle tele-operation with collaborative control. In *Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, Washington D.C., 2002.

[35] S. Franklin and A. Graesser. *Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents.* Intelligent Agents III: Agent Theories, Architectures, and Languages. Berlin: Springer, 1997.

[36] B. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. of Robotics Research*, 23(9):939–954, 2004.

[37] B. M. Ghosh and M. G. Helander. A systems approach to task allocation of human-robot interaction in manufacturing. *Journal of Manufacturing Systems*, 5:41–49, 1986.

[38] S. Gottschlich and C. Ramos. Assembly and task planning: a taxonomy. *IEEE Robotics and Automation Magazine*, 1(3):4–12, 1994.

[39] H. Huang, E. Messina, and J. Albus. Toward a generic model for autonomy levels for unmanned systems (alfus). In *Proceedings of the 2003 PerMIS Workshop*, Gaithersburg, MD, 2003.

[40] H. Huang, E. Messina, R. Wade, R. English, and B. Novak. Autonomy measures for robots. In *Proceedings of IMECE: International Mechanical Engineering Congress*, 2004.

[41] P. Kim, B. Williams, and M Abramson. Executing reactive, model-based programs through graph-based temporal planning. In *Proceedings of ISAIRAS-01*, 2001.

[42] J. Kolecki and M. Siebert. Overview of mars system-environment interactions. In *27th AIAA Plasmadynamics and Lasers Conference*, New Orleans, LA, 1996.

[43] U. Kumar, J. Knezevic, and J. Crocker. Maintenance free operating period - an alternative measure to mtbf and failure rate for specifying reliability? *Reliability Engineering and System Safety*, 64(1):127–131, 1999.

[44] G.A. Landis. Robots and humans: Synergy in planetary exploration. In *Conference on Human Exploration, Space Technology and Applications International Forum*, Albuquerque, NM, February 2003.

[45] A. Mali. Hierarchical task-network planning as satisfiability. In *Proceedings of the European Conference on Planning (ECP)*, Durham, UK, 1999.

[46] M. J. Mataric. Learning in behavior-based multi-robot systems: policies, models and other agents. *Journal of Cognitive Systems Research*, 2:81–93, 2001.

[47] S. H. Mody and H. L. Bussey. An overview of space robotic operations principles for human spaceflight and exploration. In *1st Space Exploration Conference*, Orlando, FL, January 2005.

[48] M. Montemerlo, J. Pineau, N. Roy, and S. Thrun. Experiences with a mobile robot guide for the elderly. In *18th National Conference on Artificial Intelligence*, 2002.

[49] B. Nardi. Studying context: A comparison of activity theory, situated action models, and distributed cognition. In *Proceedings of the East-West International Conference on Human-Computer Interaction*, St. Petersburg, Russia, 1992.

[50] NASA. The vision for space exploration. Tech. Rep. NP-2004-01-334-HQ, 2004.

[51] NASA Ames Computational Science Division. *Intelligence Report, NASA AMES Computational Sciences Division*, 2004.

[52] A. L. Newton, C. Nehaniv, and K. Dautenhahn. The robot in the swarm: An investigation into agent embodiment within virtual robotic swarms. In *ECAL 2003, LNAI 2801*, pages 829–838, 2003.

[53] M. Nicolescu and M. Mataric. Learning and interaction in human-robot domains. *IEEE Transactions on Systems, Man and Cybernetics*, 31(5):419–430, 2001.

[54] Noldus. Noldus information technology. http://www.noldus.com/site/nav10000. Accessed: 4/06.

[55] C. Ntuen and E. Park. A formal method to characterize robot reliability. In *Proceedings of Annual Reliability and Maintainability Symposium*, pages 395–398, 1993.

[56] Institute of Electrical and Inc. Electronics Engineers. Ieee recommended practice for the design of reliable industrial and commercial power systems. IEEE Std 493-1997, 1998. IEEE Gold Book, 1998.

[57] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 30(3):286–297, 2000.

[58] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun. Towards robotic assistants in nursing homes: Challenges and results. *Robotic and Autonomous Systems*, 42(4):271–281, 2003.

[59] M.E. Pollack, L. Brown, and et al D. Colbry. Pearl: A mobile robotic assistant for the elderly. In *AAAI 2002 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, Edmonton, Alberta, Canada, 2002.

[60] V. Potkonjak, S. Tzafestas, and D. Kostic. Concerning the primary and secondary objectives in robot task definition. *Mathematics and Computers in Simulation*, 54(1):145–157, 2000.

[61] T. Quick and K. Dautenhahn. Making embodiment measurable. In *Proceedings of '4 Fachtagung der Gesellschaft fr Kognitionswissenschaft*, Bielefeld, Germany, 1999.

[62] G. Rodriguez and C. Weisbin. A new method for human-robot system performance. *Autonomous Robots*, 14:165–178, 2003.

[63] G. Rodriguez, C.R. Weisbin, and R.W. Easter. A new method for human-robot system resiliency evaluation. JPL Publication 02-26, 2002.

[64] J. Saleh, E. Lamassoure, and D. Hastings. Space systems flexibility provided by on-orbit servicing: Part i. *Journal of Spacecraft and Rockets*, 39(4):551–560, 2002.

[65] J. Saleh, E. Lamassoure, and D. Hastings. Space systems flexibility provided by on-orbit servicing: Part ii. *Journal of Spacecraft and Rockets*, 39(4):561–570, 2002.

[66] J. H. Saleh, D. E. Hastings, and D. J. Newman. Flexibility in system design and implications for aerospace systems. *Acta Astronautica*, 53:927–944, 2003.

[67] P.S Schenker, T.L. Huntsberger, P. Pirjanian, and E.T. Baumgartner. Planetary rover developments supporting mars exploration, sample return and future human-robotic colonization. In *Proc. 10th Conf. on Advanced Robotics*, pages 31–47, 2001.

[68] D. L. Schneider, D. Tesar, and J.W. Barnes. Development and testing of a reliability performance index for modular robotic systems. In *Proc. Annual Rel. Maintain. Symp.*, pages 263–271, Anaheim, CA, 1994.

[69] J. Scholtz. Theory and evaluation of human-robot interactions. In *Proceedings of the Hawaii International Conference on Systems Sciences*, 2003.

[70] T. B. Sheridan. *Humans and Automation: Systems Design and Research Issues.* New York: Wiley, 2002.

[71] R. Simmons and D. Apfelbaum. A task description language for robot control. In *Proc. Conf. on Intelligent Robotics and Systems (IROS)*, Vancouver, CA, 1998.

[72] D. E. Smith, J. Frank, and A. K. Jonsson. Bridging the gap between planning and scheduling. *Knowledge Engineering Review*, 15(1):47–83, 2000.

[73] MacDonald Dettwiler Space and Advanced Robotics Ltd. The shuttle remote manipulator system - the canadarm. http://www.ieee.ca/millennium/canadarm/canadarm-technical.html. Accessed: 5/05.

[74] V. Sridharan and P. Mohanavadivu. Some statistical characteristics of a repairable, standby, human and machine system. *IEEE Transactions on Reliability*, 47(4):431– 435, 1998.

[75] L. Suchman. Plans and situated action ten years on. *The Journal of the Learning Sciences*, 12(2):259–264, 2003.

[76] Phoenix Technologies. 3d real-time mocap systems. http://www.ptiphoenix.com/products/visualeyez/. Accessed: 4/06.

[77] S. Thrun. Toward a framework for human-robot interaction. *Human-Computer Interaction*, 19:9–24, 2004.

[78] K. J. Vincente. *Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-based Work.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1999.

[79] I.D. Walker and J.R. Cavallaro. The use of fault trees for the design of robots for hazardous environments. In *Proceedings Annual Reliability and Maintainability Symposium*, pages 229–235, 1996.

[80] C. Weisbin, D. Lavery, and G. Rodriguez. Robots in space: Us mission requirements in technology into the next century. *Autonomous Robots*, 4:159–173, 1997.

[81] B. Williams, P. Kim, M. Hofbaur, J. How, J. Kennell, J. Loy, R. Ragno, J. Stedl, and A. Walcottt. Model-based reactive programming of cooperative vehicles for mars exploration. In *Proceedings of ISAIRAS-01*, 2001.

[82] Y. Yamada, T. Yamamoto, T. Morizono, and Y. Umetani. Fta-based issues on securing human-safety in a human/robot coexistence system. In *Proc. IEEE Systems, Man and Cybernetics*, pages 1063–1068, 1999.

[83] L. Young and G. Briggs. Smart rotorcraft field assistants for terrestrial and planetary science. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT, 2004.