# Coactive Design

## Designing Support for Interdependence in Human-Robot Teamwork


Matthew Johnson

# Coactive Design

Designing Support for Interdependence
in Human-Robot Teamwork

# Coactive Design

## Designing Support for Interdependence in Human-Robot Teamwork

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. Ir. K.Ch.A.M. Luyben
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 30 september om 12:30 uur
door

Matthew James JOHNSON,
Master of Science in Computer Science,
Texas A&M – Corpus Christi,
geboren te New York, Verenigde Staten van Amerika.

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. C. M. Jonker

Copromoter: Dr. M. B. van Riemsdijk

Samenstelling promotiecommissie:

| | |
|---|---|
| Rector Magnificus | voorzitter |
| Prof. dr. C. M. Jonker | Technische Universiteit Delft, promotor |
| Dr. M. B. van Riemsdijk | Technische Universiteit Delft, copromotor |
| Dr. J. M. Bradshaw | Florida Institute for Machine Cognition |
| Prof. dr. E. Visser | Technische Universiteit Delft |
| Prof. dr. R. Babuska | Technische Universiteit Delft |
| Prof. dr. G. Kaminka | Bar Ilan University |
| Prof. dr. M. Goodrich | Brigham Young University |
| Prof. dr. M. Neerincx | Technische Universiteit Delft, reserve |

This research project was supported by the Dr. Morley Stone and Air Force Research Lab (AFRL), the Defense Advanced Research Projects Agency (DARPA), the Institute for Human & Machine Cognition (IHMC) and the Interactive Intelligence group at Delft University of Technology.

ISBN 978-94-6186-354-6

Cover artwork by John Carff

***People ignore design that ignores people***

**– Frank Chimero**

# Preface

“Design” is an interesting word. It is both a noun and a verb. As Coactive Design was developed, there was always a question as to which of these senses was being referred to. At times it seemed like the noun. Other times it seemed like the verb. In the end, I am convinced that both senses apply.

Design, as a verb, is about process. The purpose of the design process is to find solutions. The Coactive Design method prescribes a process to help designers of human-machine systems to find solutions. Effective performance is an important criterion, so there is a necessarily utilitarian aspect to the design process. The end product must do something and, if well designed, it should do it well. The development of a method to assure effective human-machine performance was the first objective of the work described in this thesis.

However, “design” is also a noun. Sometimes the word is used in this sense to refer to the plan or specification for some result, but it can also connote something about the style in which the plan is rendered in the product. Just as you can quickly tell whether a painting is in the Impressionist style or whether a building is inspired by Baroque, the design reveals something about the aesthetic sense of the designer, his culture, and what inspires his passion for the work.

I hope that those who use the Coactive Design method will not only develop their systems according to a particular pattern, but that a characteristic “style” in the result will be obvious to the careful observer. That style should be apparent in the way people and machines interact as part of the joint human-machine system. If the user interface is layered on the machine as an afterthought, the spirit of Coactive Design has been lost. If the operator’s role is confined to starting and stopping the activity, it does not realize the objective of Coactive Design. However, if people are engaged with machines throughout the joint activity, fully aware of the status, and able to interject and receive assistance at any point, then it begins to look like Coactive Design. If there is a place for human judgment and flexibility to support human creativity, then it is Coactive Design. If the system fits so well the machine feels not like a distant tool, but an extension of your own capabilities then it is truly Coactive Design.

## Acknowledgements

The underlying theme of Coactive Design is that all human-robot activity is interdependent activity. The same can be said of the activity that went into the creation of this thesis. It was highly interdependent with a great number of people. Many people have encouraged me, assisted me, and mentored me along the way.

I begin by thanking Jeffrey Bradshaw and Paul Feltovich from the Florida Institute of Human and Machine Cognition (IHMC). Together they helped me reach beyond my engineering foundation and see a broader picture of not just robots, but human-machine systems. Their influence is visible in both my work and my writing, and I am the better for it. Jeff was also responsible for connecting me with Catholijn Jonker, my promoter at TU Delft. Jeff and Paul were my local connection, affording me an opportunity to have more face-to-face time with mentors than would normally have been possible working remotely. As part of my advisory group, they supported and encourage me through to the end.

Another member of my advisory group that I must thank is Maarten Sierhuis. Maarten provided a critical eye. His insightful questions forced me to dig deeper in my work. Maarten played an important role in bringing clarity to my early ideas and was also instrumental in connecting me with Catholijn and enabling this PhD process.

I would also like to thank Jurriaan van Diggelen. Jurriaan visited IHMC one summer and I had the privilege of working with him. He enlightened me to the opportunities of getting a PhD outside the United States. His excellent work and good friendship instilled me with a propensity toward a PhD program in the Netherlands. He also gave me a most enjoyable tour of Utrecht.

There are many people at TU Delft that are deserving of thanks. I would like to thank the TU Delft staff and students who participated in my experiments conducted at TU Delft. Experiments require participants and demand time and effort from those who volunteer. I appreciate the effort of all the participants that contributed. I also would like to thank the administrative staff. Anita Hoogmoed was highly instrumental in coordinating all of my activities. Both Ruud de Jong and Wouter Pasman played a particularly important role in providing IT support for my experiment. Tjerk de Greef gave me some great advice as I began my journey and I thank him for his assistance in arranging things at the end. I thank Koen Hindriks for all his help with “Blocks World for Teams” and for continuing the effort in his coursework. Lastly, Maaike Harbers for kindly inviting me to various activities so I could feel like a part of the wonderful interactive intelligence group at Delft.

At IHMC, I must thank Robert Hoffman for his counsel on various issues like statistics, experimental design and conceptual development. His guidance was very valuable in the development of this thesis.

John Carff and Daniel Duran of IHMC are also deserving of much thanks. John and Daniel were integral to the development of the UAV work presented in this thesis. Both helped develop the system and worked long hours on the software and the hardware to get things working. I appreciate their dedication and efforts that went into making the project a success. They also happen to be great friends who share a passion for unmanned aerial vehicles.

The IHMC DARPA Robotics Challenge (DRC) team includes over forty people, including many Dutch members such as Twan Koolan, Jesper Smith, Tomas de Boer, Jeff van Egmond, Wessel Straatman, Riewert van Doesburgh, and Maarten Griffioen. I would like to thank all of those who worked on the Virtual Robotics Challenge presented in this thesis and those who worked on the DRC Trials. This was a once in a lifetime opportunity for everyone involved. For me personally, the timing was an amazing blessing. It provided an unprecedented opportunity to evaluate the Coactive Design methodology. Though I was a bit exacting on design issues, I appreciated everybody's tolerance on my design obsession. In particular, I would like to thank Jerry Pratt and Peter Neuhaus. I greatly appreciate their leadership in the robotics lab, their support of my work, and their friendship.

Two of the main contributors to my work remain to be thanked. Birna van Riemsdijk was the co-promoter on this thesis. Birna was immeasurably helpful in so many ways. She provided critical technical evaluation of my work, pushing it to always be better. She was also extremely invaluable in coordinating all the aspects of the PhD process. She helped me with administrative forms, she helped organize and run the experiments, she coordinated everyone's calendars, she helped with all of the papers, she helped coordinate the thesis and bring it to its finished state. I cannot thank her enough for everything she did.

Finally, I must thank my promoter Catholijn Jonker. Without Catholijn, none of this would have been possible. I truly appreciate her willingness to take me on as a PhD student and her support throughout the process even though I was remote in Pensacola. I thank her (and Birna) for the many Skype sessions that allowed me to stay in touch from across an ocean. Catholijn is an ideal promoter. She has a knack for finding the right balance between pushing you to do your best and encouraging you to hang in when things are difficult. She has been a fantastic mentor and I could not have wished for a better promoter.

In closing I would like one more time to thank my advisory group of Catholijn, Birna, Jeff, Paul, and Maarten. I have expressed appreciation for their academic support, but more importantly I would like to let them know how much I appreciate

their friendship. While I have heard stories of bad experiences during the PhD process, I have none of my own to contribute. These five people are the reason behind this.

I would be remiss to not thank those “behind the scenes” that give me the strength to do my work. I thank my parents for providing a loving and supportive childhood and for instilling me with the belief that you can do anything you set your mind to. I thank my daughters, Anna and Gracie. They have both grown in to beautiful young ladies. As a father, I could not be prouder of the wonderful character each possesses. Their support during this process was so important to me. Finally, I must thank Rachel. My wife, my confidant, my best friend, and my love. You have supported me in the pursuit of all of my dreams. Without you, I would be lost. Elephant shoes.

Matthew Johnson

Delft, 2014

# Table of Contents

***The scientist is not a person who gives the right answers, he's one who asks the right questions.***

**– Claude Lévi-Strauss**

# 1 Executive Summary[1]

## 1.1 Motivation

This thesis is born of the frustration of designing "autonomous systems" within our research team and from observing the "autonomous systems" designed and built by others. These systems tend to be frail and have difficulty doing their work in real world settings fraught with uncertainty, ambiguity, and surprise. As evidence, consider the 2013 United States Department of Defense (DoD) Unmanned Systems Integrated Roadmap which states "Nearly all unmanned systems require active control of basic vehicle operations and behavior that affects communications, manpower, and system effectiveness" (2013, p. 29). The main reason for this is identified in the Defense Science Board's (DSB) assessment of the role of autonomy where they state, "the experience with autonomous systems is that they are often brittle when used in new ways" (2012, p. 58). What we desire is for these systems to be more effective. More specifically, we desire them to be resilient. Resilience is not about optimal behavior, it is about survival and mission completion. Resilience is the ability to recover from or adjust easily to misfortune or change.[2] David Woods describes it this way: "Resilience then concerns the ability to recognize and adapt to handle unanticipated perturbations that call into question the model of competence, and demand a shift of processes, strategies and coordination" (2006, p. 22). His description captures the two essential components of resilience: recognition of problems and flexible alternatives to address them. So how does one design a resilient system?

The traditional approach to resilience is to improve "autonomy." For example, the DoD Unmanned Systems Roadmap states "The Department will pursue greater autonomy in order to improve the ability of unmanned systems to operate

[1] This thesis is based on the following work:
ESAW workshop - (Johnson et al., 2009)
COIN workshop - (Johnson et al., 2011)
IEEE Intelligent Systems - (Johnson et al., 2011)
IEEE Intelligent Systems - (Johnson et al., 2012)
IMAV conference - (Johnson et al., 2012)
JHRI - (Johnson et al., 2014)

[2] "Resilience." http://www.merriam-webster.com/dictionary/resilience (accessed on 1 Feb. 2014).

independently, either individually or collaboratively, to execute complex missions in a dynamic environment" (2007, p. 1). Engineering better autonomous capabilities is undeniably valuable. However, historically autonomy has been bad at recognizing problems (Feltovich, Bradshaw, Clancey, Johnson, & Bunch, 2008; Woods & Branlat, 2010) and has offered only rigid alternatives to address them (Norman, 1990; Woods & Branlat, 2010). Additionally, autonomous capabilities are a necessary, but insufficient condition for resilience. This is because "there are no fully autonomous systems just as there are no fully autonomous soldiers, sailors, airmen or Marines. Perhaps the most important message for commanders is that all systems are supervised by humans to some degree, and the best capabilities result from the coordination and collaboration of humans and machines" (Defense Science Board, 2012, p. 24). We concur with the DSB's findings and propose human-machine teaming as an alternative approach to achieving resilience.

The view of robots as teammates has grown as the field of robotics has matured. The future will belong to collaborative or cooperative systems that do not merely *do things for* people, "autonomously," but that can also *work together with* people, enhancing human experience and productivity in everyday life (Bradshaw, Dignum, Jonker, & Sierhuis, 2012). While working together with people increases complexity as compared to standalone systems, it also brings an opportunity for extending individual capabilities and increasing resilience through teaming. Eduardo Salas et al. (Salas, Cooke, & Rosen, 2008) provide insight into why humans work in teams. Some of the reasons they state are that "teams are used when errors lead to severe consequences; when the task complexity exceeds the capacity of an individual; when the task environment is ill-defined, ambiguous, and stressful (2008, p. 540)." These reasons correlate with making a human-human team more resilient and have analogies in human-machine teams. "However, simply installing a team structure does not automatically ensure it will operate effectively. Teamwork is not an automatic consequence of co-locating people" (Baker, Day, & Salas, 2006, p. 1579). Similarly, putting a human "in-the-loop" does not guarantee effective human-machine teaming. If teaming is to be a viable alternative approach to resilience, then it will be important to understand how a developer designs a system to work effectively as a teammate.

Intuitively, effective teamwork implies coordination of activity, cooperation among participants and collaboration. However, all these terms are too abstract to give direct guidance to human-machine system designers and developers. The challenge is to translate high-level concepts such as teamwork and collaboration into specific requirements that can be implemented within control algorithms, interface elements, and behaviors. While there are plenty of textbooks on how to make a robot arm move to a specified position, there is relatively little guidance on how a human and machine can work effectively together to complete the same task. It is true that the social sciences and human factors studies have provided useful theories on human needs and capabilities. Unfortunately, this guidance often does not translate to the kind of specificity needed to engineer the requirements of an effective human-machine system (Robert R Hoffman & Deal, 2008). To address this need, we began a process of investigation to develop practical and specific guidance for designers. The result of this inquiry is an approach to human-machine design we call Coactive Design.

Prior to developing the Coactive Design perspective and methodology, we investigated the most popular approaches to human-machine design (Chapter 3.2). These approaches take an autonomy-centered perspective — a perspective that was pervasive in the training of just about every graduate student in the field. This perspective is a limiting one that often results in the kind of frustration captured in Bainbridge's article on the ironies of automation (Bainbridge, 1983). She states "the irony that one is not by automating necessarily removing the difficulties, and also the possibility that resolving them will require even greater technological ingenuity than does classic automation (1983, p. 778)." Contributing to this frustration were common misconceptions that inhibit a proper mindset for designing human-machine teams. Some misconceptions derive from several fallacies associated with autonomy (Chapter 4.3). Others derive from the pervasive concept of "levels of autonomy" (Chapter 4.2) which we, along with the DSB (2012, p. 23), argue is a construct that has outlived its usefulness. These misconceptions underscore the need for new perspectives.

So, returning to the original question, how does one design a resilient system? Coactive Design breaks with traditional approaches by focusing on effective management of the interdependencies among human-machine team members (Johnson, Bradshaw, Feltovich, Jonker, et al., 2011). Providing support for interdependence enables members of a human-machine team to recognize problems and adapt. Support for a variety of interdependence relations makes a team flexible. Flexibility, in turn, makes the team resilient by providing alternative ways to recognize and handle unexpected situations. We now turn to the specific contributions of Coactive Design toward design of resilient systems.

## 1.2 Contributions

Coactive Design, as presented in this thesis, makes five major contributions: 1) a new design perspective based on interdependence, 2) a richer understanding of interdependence, 3) a new model for human-machine systems, 4) a new design method, 5) and a new tool, called the Interdependence Analysis Table, to assist the designer in system design and analysis. Chapters 5-8 will provide the details necessary for other designers to apply this approach to their own human-machine challenges.

The first contribution of Coactive Design is a change in focus (Chapter 5). Focusing on interdependence is a clear break from the autonomy-centered perspectives that dominate current research. Coactive Design is focused on systems where the human and machine are engaged in teamwork. Besides implying that more than one party is involved, the term "coactive" is meant to convey the *type* of involvement. Consider an example of playing the same sheet of music solo versus as a duet. Although the music is the same, the processes involved are very different (Clark, 1996). The difference is that the process of a duet requires ways to support the interdependence among the players. This is a drastic shift for many autonomous robots, most of which were designed to do things as independently as possible. The term "coactive design" is about designing in a way that enables effective teamwork through support for interdependence.

The second major contribution of this work is a definition of interdependence and an understanding of the design implications of this definition (Chapter 6). The central role of interdependence demands a rich understanding of interdependence itself. In his seminal book, James D. Thompson (1967) recognized the importance of interdependence in organizational design, just as we are proposing its importance in human-machine systems. The correlation is made clear by Thompson's description of an organization as an "open-system, indeterminate and faced with uncertainty" (p. 13). He also noted that there was a lack of understanding about interdependence — something still true today. Understanding the nature of the interdependence between team members provides insight into the kinds of coordination that will be required of them. Indeed, we assert that coordination mechanisms in skilled teams arise largely because of such interdependencies. For this reason, understanding interdependence is an important requirement in designing systems that will work jointly with people. This thesis argues that managing interdependencies is the mechanism by which we achieve the higher level concepts of coordination, collaboration and teamwork.

The third major contribution is a new system model for human-machine system design (Chapter 7). We have already referred to the need to "manage" interdependencies and to "support" interdependent relationships — this chapter begins to describe how we think this may be done. Our system model highlights three key team capabilities, over and above task capabilities, that are needed for effective human-machine collaboration: observability, predictability and directability. For team members, these three capabilities enable resilience, allowing them to "recognize and adapt to handle unanticipated perturbations" (Woods & Hollnagel, 2006, p. 22). From a designer's perspective, observability, predictability, and directability are important because they provide guidance on how to identify design requirements. By determining how these capabilities must be supported in order to be capable of understanding and influencing team members, designers can create a specification. This design stance necessarily shapes not only the “user interface” for the human but also the implementation of a robot’s autonomous capabilities. The shaping process is provided by the three team capabilities in our system model which capture three of the key elements required for effective teamwork.

The Coactive Design *method* is the fourth major contribution (Chapter 8). It is a method for designers building highly interdependent systems. It provides the first step by step procedure for designing interdependent systems, based on the perspective provided by Chapter 5, the understanding provided by Chapter 6 and the specific support requirements identified in Chapter 7. We present our method within the ecology of existing methodologies and describe how it is a bridge to design a system to work effectively as a teammate.

The fifth major contribution of this thesis is the Interdependence Analysis (IA) Table (Chapter 8.1). This is a design and analysis tool to be used in conjunction with the Coactive Design method. It is a simple, visual way to enumerate the alternative ways by which combinations of team members can achieve a goal. If a system is to be resilient and deal with a demand for “a shift of processes, strategies and coordination” (Woods & Hollnagel, 2006, p. 22), there must be alternative processes, strategies and coordination. The IA Table enables designers to discover alternatives and understand how to support them in their systems. Based on the alternatives the designer chooses to support, the IA Table helps identify the independence relationships that must be supported for that relationship to be effective. This includes determining specific observability, predictability and directability requirements needed to support those relationships. Since design is always an iterative process, the IA Table supports this and helps understand the impact design changes might have on both individual and team performance.

Summarizing the contributions we answer our key question. Coactive Design is an approach that enables a developer to design a system to work effectively as a teammate. By following the Coactive Design method and using the IA Table, designers have a way to ground the high-level teamwork concepts into design specifications and requirements. These specifications are based on three key team capabilities: observability, predictability, and directability. Since the purpose of the IA Table is to identify the requirements necessary to support the desired interdependent relationships, it guides a designer to find alternatives that provide flexibility. This flexibility will add resilience to the final system.

## 1.3 Evaluation

The development of Coactive Design has matured over several years and multiple projects. Three case studies are presented in this thesis (Chapters 9-11). They capture the development of these ideas and demonstrate the applicability of Coactive Design in a variety of domains. The first case study was a simple experimental testbed that uncovered the foundational concepts of Coactive Design. The second case study demonstrates how the Coactive Design perspective produced a unique solution to a problem — one that has the characteristics of effective teamwork. It also helped solidify the method and IA Table design. The final case study is an evaluation of the full Coactive Design method in a complex real world competition.

The first case study presented (Chapter 9) is a simulated testbed that was a catalyst for maturing both the theory and the analysis technique. Prior to the development of the Coactive Design method, the IA Table, or even the definition of interdependence, we wanted to understand the problem we were attempting to address better. We knew from the literature that there were issues with autonomy yielding its expected benefit (e.g., Bainbridge, 1983; Norman, 1990; Woods & Sarter, 1997). However, we wanted to uncover what relationships exist between autonomous capabilities and performance, as well as any other influencing factors. So, we developed a joint activity testbed called Blocks World 4 Teams (BW4T). BW4T is a multiplayer game played in simulation. The game allowed for multiple human or software players in any combination. The goal of the game was for the team to find and deliver a sequence of colored boxes. There were two major results from this work. The first is the experimental results, which provide some empirical evidence against the long standing view, held by many as noted by Wickens (Hancock et al., 2013) and Blackhurst et al. (Blackhurst, Gresham, & Stone, 2011), that increased autonomy is a panacea and will improve performance while reducing cost and risk (e.g. Dempsey, 2010; Department of Defense, 2007). The results clearly show that failure to support interdependence, through observability, predictability and directability can limit the benefits anticipated by increasing autonomy. The second result is the progressive development of Coactive Design from a perspective to a method. This experiment was the catalyst for the development of the IA Table. The initial form of the IA Table was developed to understand the results. Though we anticipated the inflection in performance shown in the results, it was development of the IA Table that led to an understanding of why this change in performance occurred; failure to support interdependence. Moreover, the IA Table has proven to be a general tool usable in any domain. It has been used both to design new systems as well as analyze existing ones.

The second case study is a Florida Institute for Human & Machine Cognition (IHMC) unmanned aerial vehicle (UAV) project (Chapter 10). This project began early in the genesis of Coactive Design. Its purpose was to demonstrate how a system designed from a coactive perspective could be more resilient than a traditional one. The overall goal of the system was to demonstrate effective navigation through obstacles, which remains a challenging endeavor for current systems. This is a task that is difficult for either humans or unmanned vehicles to currently complete successfully on their own in situations of any significant complexity — harnessing the capabilities of each in effective teamwork is required. This was accomplished not by guessing at what widget or feature might be useful, but by a methodological approach to support interdependence among the human operator and the UAV through mechanisms that allowed the operator to coactively navigate. The operator could observe the internal state of the vehicle by the relative location of graphical objects. The operator could predict the resulting behavior prior to execution by a displayed path or even a virtual "fly through." Directability was supported in a variety of ways from goal specification, to waypoint modification, to obstacle correction, to state estimation adjustment. These are just a few of the ways interdependent relationships were supported to provide a lot of flexibility. The project also highlights new capabilities, impossible with most currently deployed systems, but made possible by taking a coactive design perspective on the problem. This includes things like flying from a third person view and enabling safe flight at angles orthogonal to the camera view. It is important to note that the flexibility is not a particular feature, such as allowing for graphic overlays. Flexibility is the additional options that a feature affords. The features in this UAV system demonstrate the type of flexibility a coactive system can provide.

Shifting from UAVs to complex humanoid robots, we present our third case study (Chapter 11). The DARPA Robotics Challenge (DRC) is an international competition that is like a robot Olympics. The competition consisted of three different tasks providing a broader range of activities than that of just navigation. As the design lead for IHMC's DRC entry, Coactive Design was embraced from the beginning. Extensive use of the Coactive Design method and the IA Tables was made. The competition afforded a way to evaluate this experimental approach against traditional approaches and to evaluate its ability to imbue resilience. The team's entry demonstrated a high level of resilience, placing first with a completion percentage of 86 percent; 20 percent higher than the nearest competitor. The results of applying our design approach to the DARPA Virtual Robotics Challenge (VRC) are presented as an exemplar of large scale implementation of Coactive Design. The coactive system developed was quite different from any of the other twenty-six entries in the competition. Even when considering aspects that were common across most teams, like scripted behavior, the IHMC implementation distinguishes itself by the support for interdependence. While many factors went into IHMC's first place finish, we would argue that the Coactive Design approach provided a

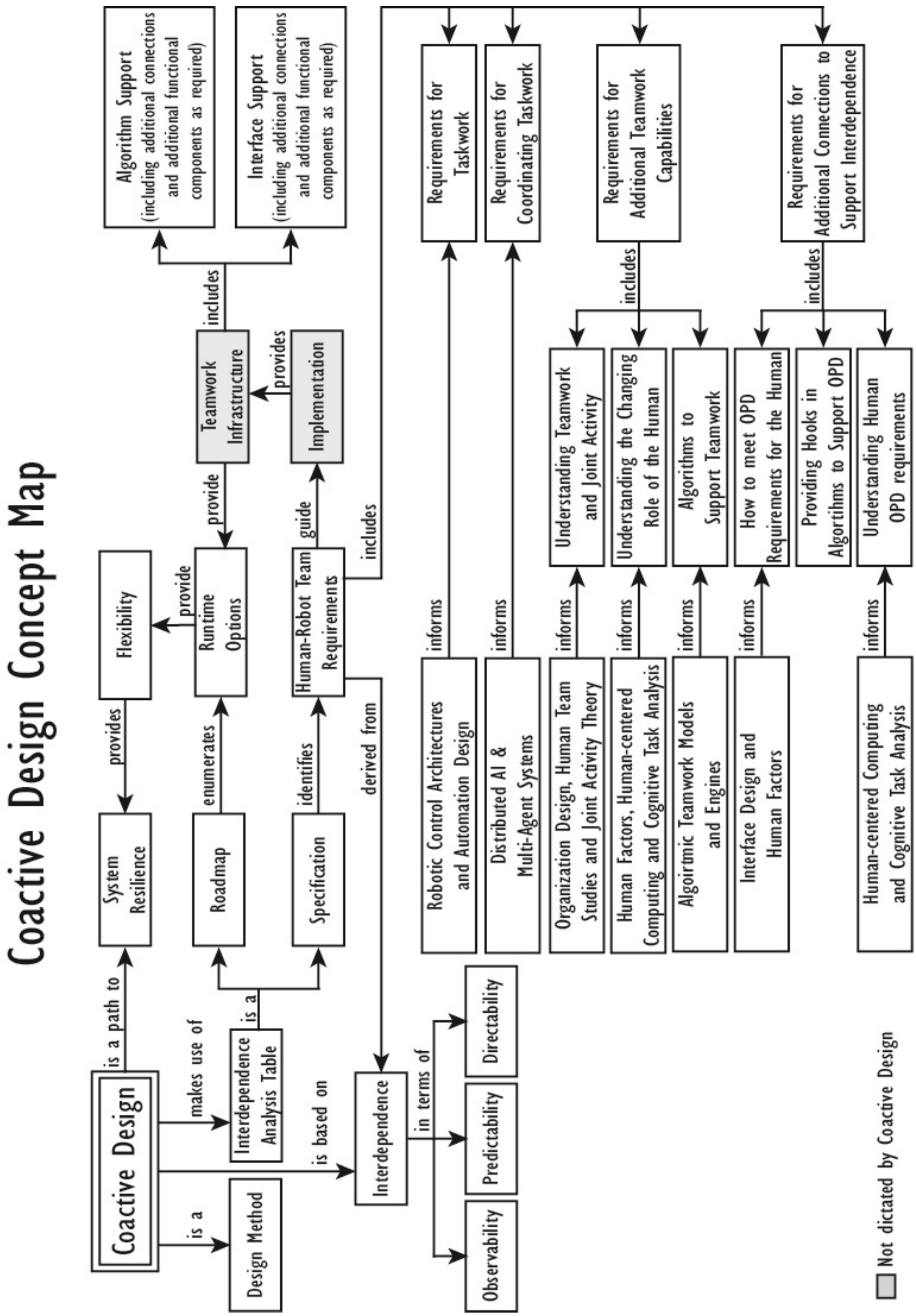



Figure 1. Coactive Design Concept Map.

distinct advantage and contributed to the success. *The success was not based on flawless performance, but on resilience in the face of uncertainty and misfortune and surprise.* As one example, consider the hose task from the VRC. The goal of the hose task was to pick up a hose from a table and attach it to a spigot. The team was required to perform five different hose tasks in which DARPA varied things like table height, hose color, and spigot location. During the five hose tasks of the VRC an average of ten "autonomous" scripts were used per run. Many teams use autonomous scripts, however, IHMC's scripts are run in a manner that supports interdependence. Only fifty percent of these were run without intervention. The team averaged nine pauses in script behavior to verify performance (i.e. observability and predictability) and seven operator corrections to scripted actions per run (i.e., directability). Even with operator intervention, eight of the fifty scripts failed to accomplish their purpose. Due to the flexibility in IHMC's system to retry, make adjustments, and use different approaches, the team was successful in recovering from all eight failures. This example is but one of many examples of resilience from the VRC experience.

## 1.4 Overview of Coactive Design

As a brief overview of Coactive Design and how it fits into the existing ecology of research areas, a concept map is provided in Figure 1. Coactive Design is a design method based on the concept of interdependence (Chapter 6). It makes use of a design tool called the Interdependence Analysis (IA) table (Chapter 8). As a design process, Coactive design produces a specification (i.e. the IA table) that details human-robot team requirements. These requirements are used to guide the implementation which provides the teamwork infrastructure. The sum of the capabilities provided by the teamwork infrastructure determines the runtime options and runtime options (in addition to how they are employed, the situation, and other factors) determine performance. In particular, the flexibility afforded by the options contributes to overall resilience of the human-robot system.

The IA table is a specification of requirements that can be used to guide implementation. However, it is unlike a traditional software specification[3] in that it does not dictate what is required. Instead the IA table is more like a roadmap of runtime options, specifically options about how the human and robot might interact. The designer is free to choose which options to support based on cost/benefit, time, the desire for flexibility or any of a myriad of factors. Each option is associated with particular capabilities and related requirements through the IA table. These requirements are described in terms of observability,

[3] Software specification definition found at http://en.wikipedia.org/wiki/Software_requirements_specification (accessed on 05JULY2014).

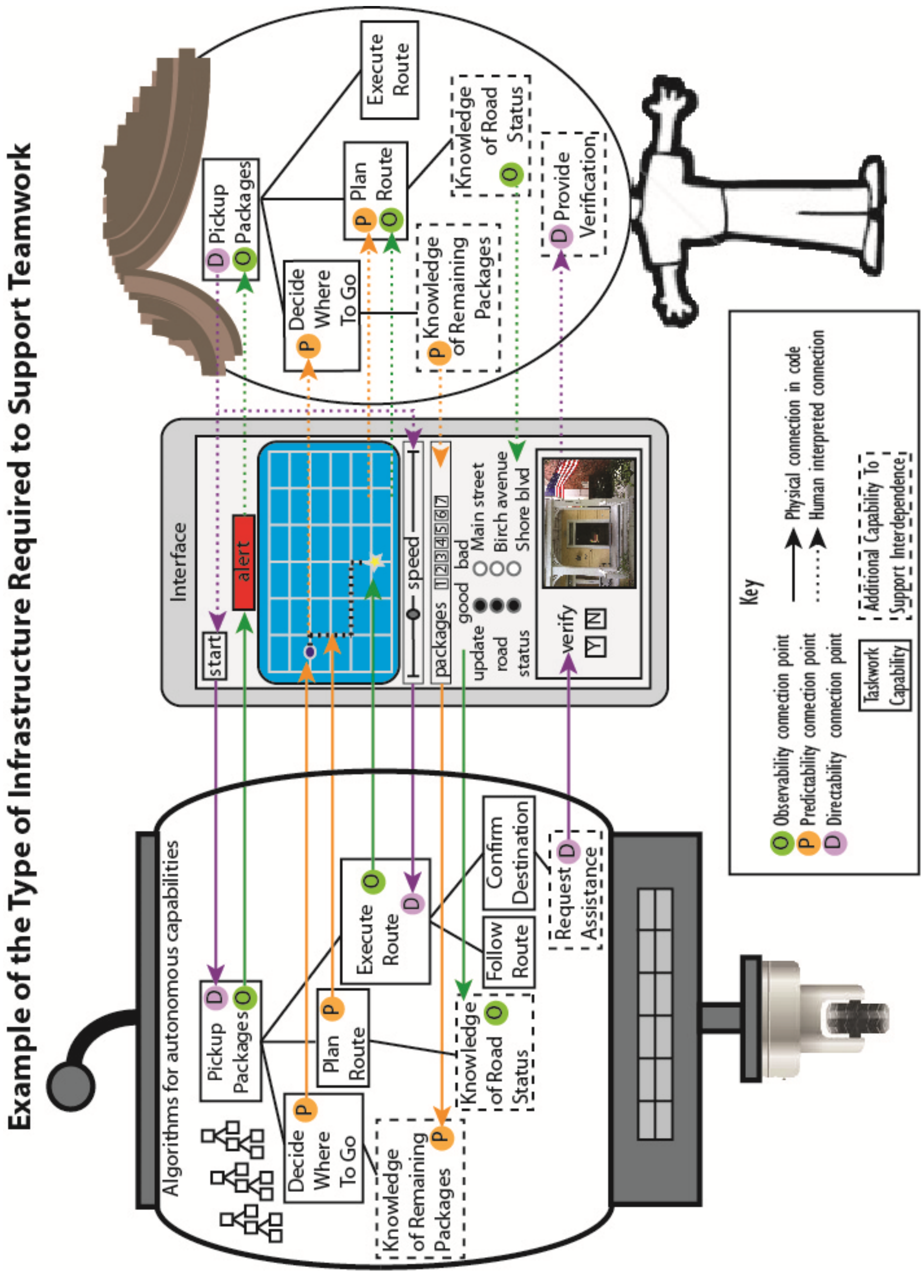



Figure 2. Example scenario of a human-robot team picking up packages together. We highlight the interdependencies (colored arrows) and the connection points required to support them. We also represent additional functional capabilities needed to support the teamwork (dashed boxes) as opposed to the required taskwork (solid boxes).

predictability and directability (Chapter 7.2.1); key elements for teamwork. Support for these elements through capabilities provides a teamwork infrastructure to support interdependent activity between the human and robot.

So what do we mean by "a teamwork infrastructure to support interdependence?" Consider, as an example, a human-robot team whose goal is to pick up packages for delivery from different locations. We will use a single robot and a single human coordinating through an interface, as shown in Figure 2. The illustration extends to multiple robots, multiple humans and interaction that is not facilitated through a physical interface (e.g. verbal and non-verbal communication), but we will keep it simple for clarity. This example and its details are contrived to illustrate our meaning of "teamwork infrastructure." Basically, we mean code modifications to both the algorithms and the interface to support the additional connections and additional functional components needed to enable teamwork.

To properly understand the requirements needed for such an infrastructure, there are many factors that must be considered. Human-robot interaction is a multi-disciplinary area and different areas of research have focused on different aspects of the problem (Chapter 3). By focusing on interdependence (Chapter 6), Coactive Design sheds some light on how these different research areas overlap and, more importantly, how they complement one another.

Robot Design typically centers on the control architecture. There are many variants, but the three most common are deliberative, reactive and a hybrid deliberative/reactive architecture (Arkin, 1998). These focus on the task work, which are the solid lined boxes inside the robot shown in Figure 2. Many of the classic design approaches in robotics (Chapter 3) focus on what to automate (Parasuraman, Sheridan, & Wickens, 2000). In other words, they help understand what solid lined boxes are needed and when the human should trigger them. Lesser used a classical goal tree to help understand distributed problem solving systems (Lesser, 1991), which is represented by the tree structures in Figure 2. These types of graph structures are commonly employed in design, particularly by those interested in teamwork (Kaminka et al., 2004; Sycara & Sukthankar, 2006; Tambe, 1997). Jennings, whose focus was distributed artificial intelligence, extended Lesser's graph formalism by adding joint goals, which he described as a more sophisticated type of coordination than simple task allocation (Jennings, 1996). Jennings emphasized the importance of interdependence, but focused on goal and resource interdependence, in line with Thompson (Thompson, 1967). In his example, if one agent had valuable information that another could use, that agent should invoke an information sharing form of cooperation (Jennings, 1996). However, neither the information, nor the capability to share information is represented on Jennings' goal tree. There is also no means by which to determine that the information was valuable.

Coactive Design addresses these limitations by including not only the tasks or goals, but also the connections needed to support joint activity and a context to evaluate the importance (see IA table in Chapter 8). Coactive Design proposes three key types of connections (Chapter 7.2.1); observability, predictability and directability (OPD). In Figure 2, task allocation is represented as a form of directability. The human directs the robot to pick up the packages for delivery. Notice that there are two endpoints, represented by colored circles in Figure 2, associated with every interdependent relationship. In this simple case, the human needs a mechanism to direct the robot, represented by the start button in the interface. The robot must have a means to trigger the appropriate algorithm for picking up packages. The connections need to be designed to work together or, as we refer to it, they must be complementary (Chapter 6.1.3). Returning to Jennings' example of information sharing, for the task in Figure 2 an example might be that the robot's battery is low. Coactive Design captures this as an observability requirement (green circle in the pickup packages behavior in Figure 2) and it would be reflected in the IA table. Jennings points out two other important aspects of interdependence. The first is that interdependencies can potentially connect any goals; whether they are close or distant in the tree. To effectively coordinate package pickup, the human could benefit from additional information, deeper than surface level task allocation. This means providing "hooks" or connection points into the existing capabilities that would not normally be exposed for fully autonomous operation. For example, the human might want to know where the robot is heading first (i.e. predictability) which requires a connection to the "Decide Where to Go" behavior. The human may want to know which route the robot will take in order to avoid conflicts (i.e. predictability) which would require a hook into the route planning component. Or the human may wish to know the current location (i.e. observability) or control the robot's speed (i.e. directability) which would require access to the route execution component. The second aspect of interdependence Jennings describes is the difference between strong and weak interdependence, which is equivalent to our use of required and optional (Chapter 6.1.4). Providing the infrastructure to permit low-level access, for example control of the robot's speed, does not mean it is required and will demand the human's attention. It simply provides additional flexibility.

Though providing connection points is essential to supporting interdependence, sometimes additional functionality beyond the original taskwork is required. In Jennings' example it was the ability to share information. In Figure 2, the dashed boxes are capabilities not required for fully autonomous operation, but essential to supporting interdependence for joint operations. For example, if the person is also picking up packages, the robot will need that information to determine what packages remain (i.e. observability) and what package the person is currently proceeding toward (i.e. predictability). The person may also have information about road status (i.e. observability) that the robot may not have access to, but could benefit from if supported. Tambe's work on STEAM and related extensions (Kaminka et al., 2004; Schurr, Okamoto, Maheswaran, Scerri, & Tambe, 2005; Tambe, 1997) provide these additional functional capabilities as a generic teamwork component. The challenge is in understanding how such rules originate, how they map to a specific context, and how the connection points are supported. Additionally, these approaches have traditionally focused on multi-agent systems and have only been applied to human-robot teams on a few occasions (e.g. Schurr et al., 2005). Fong's work on collaborative control (Fong, 2001) is particularly relevant, since it demonstrated precisely the type of additional infrastructure needed to support interdependent collaboration for an identification task. In our example, consider that the robot may have difficulty confirming it has arrived at the correct destination. Maybe the address is not prominently displayed. In Figure 2, the robot's "Request Assistance" capability represents an example of new functionality that is specific to supporting collaboration.

So far we have been focused on the robot. However, a significant amount of research has focused on the human side of the issue and the role and impact of automation. In order to support the robot's need for assistance in verifying it has reached the correct destination, there is a complementary requirement (Chapter 6.1.3) on the human side for a new role that the human must play; verification of arriving at the correct destination. This is also an example of how automation can change the nature of the interactions in the system (Christoffersen & Woods, 2002). In this case, the human would now have an additional role; to be available to provide assistance. Though we do not get access to the "code" inside a human, it is important to understand the requirements for interdependence that enable the human to be an effective part of the system. Areas of research such as human factors (e.g. Fitts & Posner, 1967; Fitts, 1951), human-centered computing (e.g. R. R Hoffman, Ford, & W, 2000; Kidd, 1992), and cognitive task analysis (e.g. Adams et al., 2009; Schraagen, Chipman, & Shalin, 2009) all provide valuable insight into these requirements. They help identify the OPD requirements (Chapter 8), depicted as colored circles inside the head of the human in Figure 2.

We now turn our attention to the interface in the center of Figure 2. The interface facilitates the interdependence relationship between the human and the robot. Here we have depicted it as a physical graphical component, but it could be accomplished through verbal or non-verbal communication or any other technique. The challenge is to take data from both the human and the robot and translate it into context relevant information usable by the other. The field of interface design provides useful guidance in proper ways to convey information to the human (i.e.

the dotted arrows in Figure 2). For example, the GEDIS guidelines have demonstrated usefulness in improving UAV displays (Lorite, Muñoz, Tornero, Ponsa, & Pastor, 2013; Ponsa & Díaz, 2007). These approaches focus on the human factors issues when designing a display. Others have used models of cognition to guide interface design (e.g. Goodrich, 2004). The key to successful interface design is about understanding both the algorithmic requirements and the human requirements, in other words, the connection endpoints in the interdependence relationship. It is not enough to know you need, for example, video data. It is important to also know that understanding that data benefits from additional context (e.g. Cooper, 2007; Drury, Richer, & Rackliffe, 2006).

Our example was chosen to demonstrate the symmetry of our design process. Some research has focused on supplementing robot limitations (e.g. Fong, 2001; Michaud et al., 2010), other work focuses on human limitations (e.g. Cooper, 2007), but both are important. Equally important as their limitations are their capabilities. Though people and robots are asymmetric in their capabilities, the fundamental mechanisms for coordination are the same. Specifically, they both require observability, predictability and directability to work with others effectively. Our example includes each party observing one another, each party needing to be able to predict some aspect of the other's action and each party directing the other in some manner. Understanding this will help a designer maximize the flexibility in their system by considering all the alternatives (e.g. Chapter 11).

In summary, we hope this simple example provides insight into the complexity of teamwork. Effective teamwork requires an infrastructure to support interdependence; appropriate connection points and additional capability beyond taskwork. This infrastructure derives from the OPD requirements of interdependence relationships. These relationships must be complementary (i.e. matching endpoints). That means the algorithms and the interface cannot be designed separately, a sentiment with growing support (Adams et al., 2009; Cooper, 2007; J. W. Crandall, Goodrich, Olsen, & Nielsen, 2005; Macbeth, Cummings, Bertuccelli, & Surana, 2012). When considering OPD requirements, consider both the perspective of the human and the robot. Remember, interdependence can be required or optional. At design time, the designer can choose which aspects of the potential infrastructure to support, based on an assessment of time, effort and utility. At runtime, relationships can be employed as needed to accomplish the work. They may be dynamically adjusted to increase situation awareness, reduce workload, or increase control as deemed appropriate. However, failure to implement support of a given aspect of the infrastructure will mean it is not available and will reduce flexibility. The infrastructure we describe helps understand how important team behaviors such as monitoring, progress appraisal and requesting assistance (Smith-Jentsch, Zeisig, Acton, & McPherson, 1998) can be understood in terms of the capabilities of the algorithms, the interface and how they relate to the role of the human.

It is our hope that designers of human-machine systems will find the Coactive Design perspective a refreshing one that sheds new light on their design challenges. It is also our hope that the methodology and tools we have presented will be valuable additions to their design processes. Coactive Design helps translate high-level teamwork concepts into reusable control algorithms, interface elements, and behaviors that enable robots to fulfill their envisioned role as teammates. Interdependence is important because it is the basis for understanding complex systems. The ways in which a designer supports interdependence in a human-machine system is the creative medium of the designer and the path by which we can add not only capability, but also flexibility and resilience to a system.

***If we knew what it was we were doing,***
***it would not be called research, would it?***

**– Albert Einstein**

# 2 Introduction

## 2.1 Problem Statement

Robots hold a special place in the imagination of humankind. The view of a personal robotic helper is as old as the concept of a robot itself. For a long time though, robots have been relegated to perform repetitive predetermined tasks in isolation, separate from both humanity and the real world. Recent successes in transitioning robots from their protective settings to the real world come with a caveat. These systems tend to be frail and have difficulty doing their work in real world settings fraught with uncertainty, ambiguity, and surprise. As evidence, consider the 2013 Unmanned Systems Integrated Roadmap which states "Nearly all unmanned systems require active control of basic vehicle operations and behavior that affects communications, manpower, and system effectiveness" (p. 29). The main reason for this is identified in the Defense Science Board's (DSB) assessment of the role of autonomy where they state, "the experience with autonomous systems is that they are often brittle when used in new ways" (Defense Science Board, 2012, p. 58). What we desire is for these systems to be more effective. More specifically, we desire them to be resilient. Resilience is the ability to recover from or adjust easily to misfortune or change.[4] David Woods and Erik Hollnagel describe it this way: "Resilience then concerns the ability to recognize and adapt to handle unanticipated perturbations that call into question the model of competence, and demand a shift of processes, strategies and coordination" (2006, p. 22). Their description captures the two essential components of robotic resilience: recognition of problems and flexible alternatives to address them. Current systems do not exhibit these characteristics, and thus the problem addressed in this thesis is:

*How does one design a resilient robotic system?*

[4] "Resilience." http://www.merriam-webster.com/dictionary/resilience (accessed 1 Feb. 2014)

## 2.2 Research Claims

The overall objective of this dissertation is to develop a process by which designers can build resilient human-robot systems. Traditional robotics approaches take an autonomy-centered perspective, focusing on how and what to automate (Parasuraman et al., 2000). Our new approach, called Coactive Design, will shift the focus from developing capabilities where machines and humans can work as independently as possible, to developing capabilities that allow them to work together – coactively. The fundamental principle that serves as the foundation for Coactive Design is that the underlying interdependence of participants in joint activity is a critical element in the design of human-machine systems.

There are two primary claims in this thesis. The first is that:

> *Interdependence is an effective basis for a design and analysis model of human-machine systems.*

This claim is the foundation of the Coactive Design approach. Interdependence is the key design element because it is the basis for understanding complex distributed systems. In particular, how a human and robot can work together as a team. As illustrated in the concept map, shown in Figure 1, interdependence is used to derive the system requirements which can be used to design a new system or analyze an existing one. Providing an infrastructure that supports interdependence is what provides runtime options. The flexibility afforded by the runtime options contributes to overall resilience of the human-robot system. The second claim of this thesis is that:

> *Resilience in human-machine systems benefits from a teamwork infrastructure designed to exploit interdependence.*

Support for this claim addresses the problem statement of how one designs a resilient system. Designing to exploit interdependence means that appropriate connections between algorithms and interface are provided as well as any machine intelligence required to leverage these connections, as described in Chapter 1.4. This teamwork infrastructure makes the system capable of resilience. However, there is still a requirement for intelligent use of the infrastructure by the human teammates. Notwithstanding, no amount of intelligence in a human teammate can compensate for a lack of supporting teamwork infrastructure designed to exploit interdependence.

## 2.3 Research Aim

In this thesis, we aim to provide a design method and guidance that will be helpful to engineers, not merely conceptual designers. Some of the best guidance currently available on designing human-robotic systems are suggestions like "determine what to automate and to what extent" (Parasuraman, Sheridan, & Wickens, 2000, p. 287) and use a "flexible interaction strategy, where each agent can contribute to the task what it does best" (Allen, Guinn, & Horvitz, 1999, p. 14). This guidance is far from the type of specification desired by an engineer trying to implement such systems. Thus our aim is:

> *To develop a design process for identifying and exploiting interdependence in human-machine systems, in order to provide ways to recognize problems and create alternatives to address them.*

## 2.4 Research Approach

We employed three main methodologies during this investigation: theory development, simulated testbed experimentation, and validation through case studies on complex real robotics systems. Each method contributed insights and there was significant overlap and iteration between the methodologies throughout the process of investigation.

In order to develop a new design theory, a thorough review of existing literature was conducted. The purpose of the review was to understand the limitations of existing theories to ensure that the theory developed in this thesis is indeed novel and that its contribution adds value to the scientific community. With each testbed experiment and case study the theory was refined.

Teamwork is a complex domain and developing a controlled testbed for experimentation is challenging. It is particularly difficult to design something simple enough to allow a detailed analysis, yet complex enough to demonstrate interesting teamwork behavior. The testbed experimentation was used to further the theory and partially address the research claims.

Complex case studies were an essential part of this work. For a design process to be of value, it must address the complexities of real world human-machine systems. This means an actual person working with a physical robot doing real work in a real world environment. While a creative designer could develop a "feature" or a "widget" to solve a particular problem, the goal of this work is to provide a process by which a designer can address *any* teamwork challenge.

## 2.5 Scope

Like all work, this work is scoped to a set of particulars. This thesis is specifically focused on teamwork in human-machine systems where:

- All activity participants, both human and machine, are expected (by design) to play the role of team members. We are not focusing on, for example, an automated checkout clerk where the human has no understanding of the system other than projected human analogies. The human is expected to be familiar with the machine and the machine can assume a certain competence level from the human.
- There is an overarching activity (team goal) that makes the work joint. It should not be working solely on separate problems in the same location, for example, driving in traffic.
- The activity is long term activity, not brief encounters between a human and a machine. The work should be complex with unexpected events. It should not be, for example, a short transactional behavior like providing correct change or giving directions unless in the context of a broader activity.
- All team members are trying their best. We are specifically not addressing motivation of the humans involved and expect best effort.

When we speak of interdependence, it is specifically in the context of joint activity based on these particulars. For more discussion on the specifics of joint activity see Klein et al. (Klein, Feltovich, Bradshaw, & Woods, 2005). It is expected that some of the results could be applied outside of this scope (e.g. human-human teams, competitive instead of cooperative teams, etc.) but this is not the focus of this work. We are specifically focusing on *collaborative* or *cooperative* systems that do not merely *do things for people*, but also can *work together with people*.

***If I have seen further it is by standing on the shoulders of Giants***

**– Isaac Newton**

# 3 Background and Related Work[5]

The domain of human-robot systems crosses many disciplines from engineering and computer science to cognitive psychology and joint activity theory. Several communities have emerged specifically to address the issues of man and machine such as human-robot interaction and human-computer interaction. This section will present relevant related work and discuss how it influenced this thesis.

Traditional robotics approaches take an autonomy-centered perspective. Since autonomy is such a prevalent part of much of the prior robotics work, an overview of different usages of the term *autonomy* in the agent and robot literature will be provided in Section 3.1. We will then present prior work in the field of robotics in Section 3.2, explaining our characterization of these systems as autonomy-centered. This work has been influenced by several fields outside of robotics and so in Section 3.3 we provide a discussion of several of these areas and how they were influential.

## 3.1 Autonomy

Autonomy has two basic senses in everyday usage. The first sense, self-sufficiency, is about the degree to which an entity is able to take care of itself. Bradshaw (Bradshaw, Feltovich, et al., 2004) refers to this as the *descriptive dimension* of autonomy. Similarly, Castelfranchi (Castelfranchi, 2000) referred to this as one of the two aspects of *social autonomy* that he called *independence*. People usually consider robot autonomy in this sense in relation to a particular task. For example, a robot may be able to navigate autonomously, but only in an office environment. The second sense refers to the quality of self-directedness, or the degree of freedom from outside constraints (whether social or environmental), which Bradshaw calls the *prescriptive dimension* of autonomy. Castelfranchi referred to this as autonomy of delegation and considered it another form of *social autonomy*. For robots, this usually means freedom from human input or intervention during a particular task. To avoid the ambiguity often found in the literature, we will use the terms *self-sufficiency* and *self-directedness* in our discussion.

[5] This chapter is adapted from (Johnson, et al., 2011).

## 3.2 Prior related work in the field of robotics[6]

Within the field of robotics, there are several approaches to human-machine system design. A historical review of the dominant views of the field is presented here, though they all remain pertinent today. In fact, Supervisory Control (Sheridan & Verplank, 1978; Sheridan, 1992), one of the earliest approaches, arguably remains the most dominant perspective today.

### *3.2.1 Function Allocation and Supervisory Control*

The concept of automation—which began with the straightforward objective of replacing whenever feasible any task currently performed by a human with a machine that could do the same task better, faster, or cheaper—became one of the first issues to attract the notice of early human factors researchers. These researchers attempted to systematically characterize the general strengths and weaknesses of humans and machines (Fitts, 1951). The resulting discipline of *Function Allocation* aimed to provide a rational means of determining which system-level functions should be carried out by humans and which by machines. Thomas Sheridan and William Verplank proposed the concept of *Supervisory Control* (Sheridan & Verplank, 1978), in which people allocate tasks to one or more machines and then monitor their performance. For these types of approaches which employ task decomposition and allocation, the designer's job is to determine what needs to be done and then provide the agent or robot the capability (i.e., self-sufficiency) to do it. One of the challenges of such approaches is that the suitability of a particular human or machine to take on a particular task may vary by time and over different situations.

### *3.2.2 Adaptive, Sliding, or Adjustable Autonomy*

To address requirements for variable task allocation in different situations, there has been interest in more dynamic approaches. Gregory Dorais and David Kortenkamp (2001) define "adjustable autonomy" as "the ability of autonomous systems to operate with dynamically varying levels of independence, intelligence and control." Bernardine Dias et al. (2008) uses a similar definition for the term "sliding autonomy." Kaber and Riley (1999) define adaptive automation as "a form of automation that allows for dynamic changes in control function allocations between a machine and human operator based on states of the collective human–machine system." Sheridan (2011) discusses "adaptive automation," in which the system must decide at runtime which functions to automate and to what extent. We will use the term *adjustable autonomy* as a catch-all to refer to this concept, namely, a change in agent autonomy—in this case the self-directedness aspect—to some appropriate level, based on the situation. In each case, the system must decide

[6] Parts of Section 3.2 are adapted from (Bradshaw, et al. 2004).

at runtime which functions to automate and to what level of autonomy (Parasuraman et al., 2000). The insight these approaches provide is a need for flexibility and adaptability in a system. Our extensive work in this area (Bradshaw, Feltovich, et al., 2004; Bradshaw et al., 2005, 2008, 2003; Sierhuis, Bradshaw, Acquisti, Hoof, & Jeffers, 2003) has given us a deep understanding of the challenges associated with these approaches (Johnson, Bradshaw, et al., 2012). The main challenge is identifying what to adjust and when to adjust it. This challenge is further complicated by the difficulty of predicting the impact a change may have on the system as a whole in a given context. March, Simon, and Guetzkow point out that "one peculiar characteristic of the assignment problem…is that, if taken literally, problems of coordination are eliminated" (1993, p. 44). This is because approaches based on allocation unrealistically tend to ignore what March et al. describe as "the contingent character of activities" (1993, p. 46). Any significant form of collaboration cannot be fully addressed through mere task decomposition and allocation. It is the joint nature of key tasks that defines the heart of collaborative activity—and it is the effective management of interdependence that makes such work possible. Therefore, effective management of systems with autonomy requires an understanding of the impact a change in autonomy may have on the interdependence in the human-machine system.

### *3.2.3 Mixed-Initiative Interaction*

Though the mixed-initiative interaction approach evolved from the human-computer interaction and multi-agent systems research community, it has permeated into the robotics field because it shares similar ideas and assumptions. James Allen defines mixed-initiative as "a flexible interaction strategy, where each agent can contribute to the task what it does best" (1999, p. 14). In Allen's work, the system is able to reason about which party should initiate action with respect to a given task or communicative exchange. In a similar vein, Karen Myers and David Morley (2001) describe a framework called "Taskable Reactive Agent Communities that supports the directability of a team of agents by a human supervisor by modifying task guidance." Directability, or more accurately, task allocation is once again the central feature of the approach. Robin Murphy et al. (2000) also uses the term "mixed-initiative" to describe their attention-directing robotic system. The goal of the system was to get the human to assume responsibility for a task when a robot fails. Mixed-initiative interaction implementations have deviated from the initial goal of a "flexible interaction strategy" and tend to focus on task assignment or the authority to act. As such, the design challenge is how to vary self-directedness. The original concept of Mixed-initiative interaction contributes the valuable insight that joint activity is about interaction and negotiation, and that dynamic shifts in control may be useful.

### 3.2.4 Shared Control

An early approach that broke from the traditional task assignment view was Shared Control (Sheridan, 1992). Originally it was described as acting as "a supervisory with respect to control of some variables and direct controller with respect to other variables (Sheridan, 1992, p. 3)." More recently it has been described as Blended Shared Control where inputs combined through some functional relationship (Enes & Book, 2010). This has been shown to be useful in particular domains such as wheelchair control and training in robotic surgery. The challenge with this approach is developing the proper blending function. Additionally, the assistance is "blind" with neither party privy to the inputs and intentions of the other. This can lead to confusion and frustration, which may explain why some research in this area has concluded that less is more (Chipalkatty, Droge, & Egerstedt, 2013).

### 3.2.5 Collaborative Control

Collaborative Control is an approach proposed by Terry Fong (2001) that uses human-robot dialogue (i.e., queries from the robot and the subsequent presence or absence of a responses from the human), as the mechanism for adaptation. As Fong states, Collaborative Control "allows robots to benefit from human assistance during perception and cognition, and not just planning and command generation" (2001, p. 3). Collaborative Control distinguishes itself from the other approaches we have presented by introducing the idea that both parties may participate simultaneously in the same action. Here the ways in which the human operator could support the frail autonomy were used to shape the design of autonomous capabilities. The robot was designed to enable the human to provide assistance in the perceptual and cognitive parts of the task. The robotic assistance is not strictly required, so we are not merely talking about self-sufficiency. The key point is that the robotic assistance in this case is an integral part of the robot design and operation.

## 3.3 Prior related work outside the field of robotics

Many researchers in the field of robotics have reached outside of the robotics domain for inspiration. Given the human involvement in robotic systems, the natural consequent is to try to draw from the human and social sciences. We briefly mentioned some of the human factors work that influenced the early approaches of function allocation and supervisory control. Now we will present some additional relevant work outside of robotics.

### 3.3.1 Task Analysis

All of the approaches discussed above provide perspectives and some guidance, but none provide a method. The human-computer interaction community is closely related and has produced several methodologies. They developed Hierarchical Task Analysis (HTA) (Annett, 2003) as a method for identifying and decomposing complex tasks. Cognitive Task Analysis (CTA) (B. Crandall & Klein, 2006; Schraagen et al., 2009) has extended this methodology to include a representation of the knowledge and reasoning required to perform tasks. Goal-Directed Task Analysis (GDTA) (M R Endsley, Bolté, & Jones, 2003) is a type of CTA that includes situation awareness requirements. These approaches provide useful insight into task dependencies and human requirements for those dependencies. However, interdependence in a team can be due to more than just the task at hand. As task analysis techniques they tend to focus on the requirements for the taskwork and not the requirements of the teamwork.

### 3.3.2 Teamwork Studies

The “team” metaphor is often used when describing robotic systems. As such, it is logical to try to understand what is involved in human teams. There is an extensive body of work focused on understanding human teams. This domain has provided many models of teamwork. These models tend to be lists of characteristics or properties such as having clearly defined roles and effective communication (Larson & LaFasto, 1989; Salas, Dickinson, Converse, & Tannenbaum, 1992), or of high-level behaviors such as providing periodic updates and monitoring for errors (Smith-Jentsch et al., 1998). Some in the HRI community have adopted these models and even implemented them in their robotic or agent systems (e.g., Sycara, 2002).

While these types of informative guidance are of value, they have difficulty being embraced by system engineers building human-machine systems because they provide little in the way of practical guidance to designers and developers for analysis and implementation. The language, concepts, and products of those who focus on teamwork theory are often far removed from those who design and implement working systems (Robert R Hoffman & Deal, 2008).

### 3.3.3 Interdependence Theory

The critical role of interdependence demands a deep understanding of the concept of interdependence. Past research on interdependence in the social sciences includes the organizational theory work of James D. Thompson (1967) who identified three types of interdependence: pooled, sequential, and reciprocal. These types were characterized by the interaction between organizational units, specifically, in how the output of one unit may affect another unit (Thompson, 1967). These types of interdependence are relevant for human-machine design but are insufficient to cover the nuances of close collaboration of human and machine working jointly on a task. Other organizational theory work includes Thomas Malone and Kevin Crowston's (1994) interdisciplinary study of coordination. Their study provides descriptions of other types of interdependence and provides some valuable insight into the purpose of interdependence. However neither Thompson nor Malone defines interdependence.

Shifting to Social Psychology, John Thibaut and Harold Kelly (1959) introduced a Theory of Interdependence. It provides a description of interdependence; however, the description is limited to how an individual's behavior affects the outcomes of contingent relationships. While this type of interdependence is important in human relationships, it is insufficient for designing human-machine systems.

### *3.3.4 Joint Activity*

Another domain that has relevant theory is that of joint activity theory (Bradshaw, Feltovich, & Johnson, 2011; Klein et al., 2005; Klein, Woods, Bradshaw, Hoffman, & Feltovich, 2004), which is a generalization of Herbert Clark's (1996) work in linguistics. Our sense of joint activity parallels that of Clark (1996), who has described what happens in situations when what one party does depends on what another party does (and vice-versa) over a sustained sequence of actions. In such joint activity, we say that team members are "interdependent" (Feltovich, Bradshaw, Clancey, & Johnson, 2007).

Some of the ideas from joint activity theory have permeated into the world of robotics. For example, Cynthia Breazeal et al.'s (Breazeal, Gray, Hoffman, & Berlin, 2004) work is based on the joint activity work of Philip Cohen and Hector Levesque (1991). However her work starts with the question "What characteristics must a robot have to work effectively with its human collaborator" (Breazeal et al., 2004, p. 552). While an excellent demonstration of the characteristics, it is unclear if these characteristics are required in all systems and in all circumstances. Similar to the teamwork studies, this is not so much a method to follow as an example implementation in a specific domain.

### *3.3.5 Multi-Agent Systems*

Multi-agent systems (MAS) face many of the challenges as human-machine systems. MAS are also related to Distributed Artificial Intelligence (DAI). All of these domains need to coordinate multiple parties and deal with decentralized control. Nick Jennings (1996) work on DAI is particularly relevant. His central hypothesis is "all coordination mechanisms can ultimately be reduced to commitments and their associated (social) conventions" (1996, p. 190). This is similar to our focus on interdependence relationships. The main drawback of this work is the exclusive focus on goals. Interdependence involves more than goal dependencies. Jennings points out future needs that correlate well with the aim of this thesis:

- Produce a finer grain classification of the types of goal interdependency
- Provide a methodology for designing appropriate conventions and social conventions

Another MAS example is Milind Tambe's work on STEAM and related extensions (Kaminka et al., 2004; Schurr et al., 2005; Tambe, 1997). Tambe referred to STEAM as an "implemented model of teamwork" (1997, p.83). The work showed the value of separating the teamwork rules from the taskwork by showing reuse of teamwork rules in multiple domains. The challenge is in understanding how such rules originate, how they map to a specific context, and how the connection points are supported, as described in Chapter 1.4. Additionally, these approaches have traditionally focused on multi-agent systems and have only been applied to human-robot teams on a few occasions (e.g. Schurr et al., 2005).

### *3.3.6 Human-centered computing*

The principles of human-centered computing (HCC) and the challenges identified by that community (Robert R Hoffman, Hayes, Ford, & Bradshaw, 2012) are highly relevent to this work and have been very influential. In some sense, much of the work in this area has been performing the analysis on existing human-machine systems. This analysis has highlighted the flaws for current approaches, like those in the section on the robotics field. While providing excellent guidance to a designer, the work has not provided a methodology that provides specifications suitable for building resilient robotic systems.

## 3.4 Mapping prior work to Coactive Design

In reviewing prior work it is not feasible to provide an exhaustive review, however we have tried to provide a sufficiently broad view of related work. Within each related area of research, we have tried to highlight the insights provided by the work as well as any limitations. In Chapter 1.4 we provided an overview of Coactive Design and how we see it fitting within the ecology of existing research areas. It is important to note that the Coactive Design process is not the only path to a good solution. There are many examples across the research areas we have covered that have notable successes. Table 1 is a short sample of interesting solutions and how they map to our key requirements of observability, predictability and directability (Chapter 7.2.1). This table lists some of the many different technological features have been tried with varying degrees of success. Examples include using augmented or virtual reality, embedding video in map displays, use of the third person perspective, use of physical icons for control, and voice control. Goodrich reminds us that "the fundamental purpose of HRI is to allow a human to accomplish a task in the world (and not to allow the human to interact with a robot) (Goodrich, 2004)." Similarly, any technology feature should be evaluated based on this same criteria. How well they support observability, predictability and directability is a starting point for this evaluation process.

Table 1 Mapping prior work to key OPD requirements

| Project | Feature | O | P | D | Reference |
|---|---|---|---|---|---|
| Ground vehicle teleoperation | 3D interface and use of third person view | ✓ | ✓ | | (Nielsen, Goodrich, & Ricks, 2007) |
| Human navigation | Augmented Reality navigational cues | ✓ | | ✓ | (Reitmayr & Schmalstieg, 2004) |
| Airborne search | Video embedded in virtual map | ✓ | | | (Drury et al., 2006) |
| Ground vehicle navigation | Using augmented reality to build topographic map | ✓ | ✓ | ✓ | (Giesler, Salb, Steinhaus, & Dillmann, 2004) |
| Ground vehicle navigation | Blend control via algorithmic combination of inputs | | | ✓ | (Enes & Book, 2010) |
| Ground vehicle navigation | 3D augmented reality and embedded video | ✓ | | | (Michaud et al., 2010) |
| Aerial mission supervision | SA display and health display that allows user to adjust acceptability criteria | ✓ | ✓ | ✓ | (Macbeth et al., 2012) |
| Air vehicle mission | 3D Flight simulation to understand state | ✓ | | | (Cervin, Mills, & Wünsche, 2004) |
| Air vehicle control | Various techniques including mixed-reality, voice and physical icons | ✓ | ✓ | ✓ | (Quigley, Goodrich, & Beard, 2004) |
| Multi-air vehicle control | Overhead view and real-time mission adjustment | ✓ | | ✓ | (Brisset & Hattenberger, 2008) |

***Almost every significant breakthrough in the field of scientific endeavor is first a break with tradition, with old ways of thinking, with old paradigms.***

**– Stephen R. Covey**

# 4 The Fallacy of Autonomy[7]

Many approaches to designing more team-like cooperation between humans and machines have been proposed, including function allocation, supervisory control, adaptive automation, dynamic task allocation, adjustable autonomy, mixed-initiative interaction—most recently regrouped under the rubric of cooperative robotics. All these approaches rely on the *levels of autonomy* concept as the benchmark for machine performance and the criterion for decisions about human-machine task allocation and the supervisory control regimen. There are two theoretical issues with this concept that must be addressed. First, we argue that the concept of *levels of autonomy* is incomplete and insufficient as a model for designing complex human-machine teams, largely because it does not sufficiently account for the interdependence among team members. Second, we explore some misconceptions surrounding the topic of *autonomy* itself, which we refer to as myths of autonomy.

## 4.1 Pervasiveness of the Levels of Autonomy Concept

The concept of levels of autonomy is usually attributed to the pioneering work of Thomas Sheridan and William Verplank (1978). Their ideas were derived from a teleoperation study with underwater robots. Although the original 1978 work is often cited, the original three page table is usually condensed and simplified as shown in Table 2.

The "levels" were used to describe the space of design options, as they saw them. They range from tedious and error-prone manual operation, where humans are required to do everything (level 1), to fully autonomous operations, where the machine can perform the entire task without assistance or direction (level 10). Sheridan and Verplank realized the unlikelihood of achieving a completely autonomous solution because they "simply [did] not have available at [that] time such devices or the understanding to build such devices" (p. 1-10) for their demanding environment. Given this realization, they suggested two things:

[7] This chapter is adapted from two papers: (Johnson, et al., 2011) and (Bradshaw, et al. 2013)

- levels of automation as a means to gain some of the benefits of autonomy while not requiring a fully autonomous solution and
- supervisory control, in which humans allocate tasks to one or more machines and then monitor them.

The classic Sheridan-Verplank levels are widely cited and have had a significant impact on the outlook of robot designers. A recent survey of human-robot interaction observed that “perhaps the most strongly human-centered application of the concept of autonomy is in the notion of level of autonomy” (Goodrich and Schultz, 2007, p. 217). This seems counterintuitive. Why should the independence of a given robotic partner play a more dominant role in human-centered design of joint activity than the interdependence among the set of human-robotic team members?

Table 2. Levels of automation.*

| Level | Description |
| --- | --- |
| **High** | 10. The computer decides everything, acts autonomously, ignoring the human. |
| | 9. The computer informs the human only if it, the computer, decides to. |
| | 8. The computer informs the human only if asked, or |
| | 7. The computer executes automatically, then necessarily informs the human, and |
| | 6. The computer allows the human a restricted time to veto before automatic execution, or |
| | 5. The computer executes that suggestion if the human approves, or |
| | 4. The computer suggests one alternative |
| | 3. The computer narrows the selection down to a few, or |
| | 2. The computer offers a complete set of decision/action alternatives, or |
| **Low** | 1. The computer offers no assistance; the human must take all decisions and actions. |

* Adapted from an earlier work (Parasuraman et al., 2000)

## 4.2 Problems with the Levels of Autonomy Concept

Significant nuances in the original Sheridan-Verplank work have been forgotten through frequent use of the simplified list shown in Table 2. As a basis for our discussion, Figure 3 illustrates the richer detail in the original work. In this excerpt from the complete model, we have altered Sheridan’s level 6 by adding the tell functions and associated text from level 8. We did this to incorporate all the basic elements in a single level for discussion purposes, but it does not significantly alter the original intention because the original table had a footnote indicating other possible variations.

| Description of interaction | Human function | Computer function |
|---|---|---|
| The computer selects action, informs the human in plenty of time to stop it, and tells the human what it did only if the human asks. | Request options | |
| | | Get options |
| | Request select action | |
| | | Select action |
| | Approve action | |
| | | Start action |
| | Request tell | |
| | | Tell |

Figure 3. Altered excerpt of Sheridan-Verplank's level 6 automation. Our goal was to incorporate all the basic elements in a single level for discussion purposes and more clearly show that two parties (computer and human) are involved in the activity. The solid arrows depict hard constraints that enable or prevent the possibility of an activity. The dashed arrow indicates soft interdependence, which includes optional commands. (Adapted from Sheridan & Verplank, 1978)

The first column is the description that corresponds to an item on the simplified version of the list from Raja Parasuraman, Thomas Sheridan, and Christopher Wickens. The second column represents the human functions in the activity and the third represents the functions the computer performs. Interestingly, arrows were used between the second and third columns in the original work creating a small causal diagrams. This representation more clearly shows that two parties are involved in the activity, as opposed to the list in Table 2, which focuses solely on the computer. Additionally, these arrows represent a workflow with dependencies connecting the functions. Insightfully, Sheridan and Verplank understood that even their original richer description had limitations and stated that "as computer control and artificial intelligence become more sophisticated, certain human functions in teleoperation may be replaced, but greater need and demand will be placed upon other human functions, and in these respects the need for improved man-computer interaction will increase, not diminish" (1978, p. 1-10).

With this in mind, we have outlined several problems with the simplified concept of levels of autonomy as it is usually formulated.

### *4.2.1 Problem 1: Functional Differences Matter*

There are significant differences between performing an action and making a decision as well as between different kinds of actions. Sheridan and Verplank's original work provided a table of behavior elements that can be used to characterize a system. Their list included request options, get options, select action, approve action, start action, and tell functions. In this regard, the original levels model mixes apples and oranges—task work and teamwork. For example, in their level 1, the human handles the entire task without automation by performing the get options, select action, and start action functions. These are task-work components. On the other hand, the request options, approve action, and tell elements engage both parties in a simple form of teamwork.

The model also mixes reasoning (get options), decisions (select action), and actions (start action). Moreover, the entire approach reinforces the erroneous notion that "automation activities simply can be substituted for human activities without otherwise affecting the operation of the system" (Christoffersen and Woods, 2002).

Parasuraman, Sheridan, and Wickens' (2000) work attempted to address some of these problems by associating activity types with the 10 levels. They proposed four types (acquisition, analysis, decision, and action), but this merely highlights the importance of functional differences between the elements and ignores the issues of interdependence relating to such activities.

### *4.2.2 Problem 2: Levels Are Neither Ordinal nor Representative of Value*

Another problem is that the term *level* implies an ordinal relationship. Authors who reproduce the condensed version often add the low and high labels to levels 1 and 10, respectively, as in our Table 2. These labels imply that the levels are of increasing autonomy, but are they really? The get options function seems like a lower level of autonomy than the select option. However, if the "getter" of the options can filter the options and the receiver has no other means to know what the options are, is it really a lower level? Who holds the power in this relationship? Which has a higher value: a start action or tell? It probably depends on the criticality of what is being started and the importance of what is being told. For these and other reasons, it is more productive to think about autonomy in terms of multiple task-specific dimensions rather than in terms of a single, unidimensional scale (Bradshaw, et al., 2004).

The perspective in which we view a system can also affect our assessment of autonomy. For example, ambiguity about the term autonomy comes into play in Figure 3. Because the level shown is six out of 10, we could consider the machine semiautonomous—that is, at a mid-level of autonomy. However, with respect to the self-sufficiency perspective on autonomy, the machine could be viewed instead as fully autonomous because it can perform all aspects of the task work. On the other hand, from a self-directedness perspective, a machine functioning at this level would have no autonomy since the performance of its task work is completely subject to the direction and initiative of the human.

Our assessment of a system's autonomy also depends on the way we define the boundaries of its sphere of action. Consider the vehicles that competed in the DARPA Urban Challenge, which were designed to find their way over a given course in "fully autonomous" fashion. Although fully autonomous with respect to this one particular task, they might be far from autonomous with respect to related tasks, such as going to the store and getting groceries.

This also applies in the other direction. Several entries in the Urban Challenge were unsuccessful at completing the task but were successful at aspects of the task. For example, some could follow the road but not deal with traffic. These might be called semiautonomous, but all this term tells us is that the machine could not do everything on its own. If we redefine the task as something simpler, such as following a road without traffic, then we could once again describe the car as fully autonomous. In fact, virtually any machine could be considered fully autonomous if we define the grain size of its task to be sufficiently small. These examples make it obvious that the property of autonomy is not a mere function of the machine, but rather a relationship between the machine and a task in a given situation.

### *4.2.3 Problem 3: Autonomy is Relative to the Context of the Activity*

Autonomous capabilities are relative to the context of the task for which they were designed. When designers consider what level of autonomy is appropriate, they are assuming some level of granularity and using that to define activity boundaries. Sheridan and Verplank's original table title was "Levels of automation in man-computer decision making for a single elemental decisive step." In other words, level 10 represents full autonomy relative to the single elemental decisive step or activity. Unfortunately, over time researchers have generalized this to all activity in complex systems involving teams of humans and machines. This goes far beyond the original scope and might explain Sheridan's comment that "surprisingly, the level descriptions as published have been taken more seriously than were expected" (2000, p. 206).

Functions are not automated in isolation from task context. Therefore, when system designers automate a subtask, they are really performing a type of task distribution and, as such, have introduced novel elements of interdependence within the system. This is the lesson to be learned from studies of the *substitution* myth (Christoffersen and Woods, 2002), which states that reducing or expanding the role of automation in joint human-automation systems can change the nature of interdependent and mutually adapted activities in complex ways. To effectively exploit automation's capabilities (versus merely increasing automation), we must coordinate the task work—and the interdependence it induces among players in a given situation—as a whole.

As an example, consider the major assumption underlying the Sheridan-Verplank levels that the human, in a supervisory role, is the initiator of the activity and has an implied obligation to monitor the activity. Although this is not explicit in the model, it can be derived from the fact that the request options action is only available to the human and that the tell option is only available to the computer. Roles are not simple titles; rather they are mechanisms by which we describe capabilities and their interdependence.

### *4.2.4 Problem 4: Levels of Autonomy Encourage Reductive Thinking*

Other researchers have raised the issue of "keeping things too simple" in the design of cognitive systems (e.g., Feltovich, Hoffman and Woods, 2004). The *levels of autonomy* concept demonstrates several of these oversimplifications. Some have already been mentioned, such as ignoring functional differences, which could include treating heterogeneous elements as homogeneous and ignoring task context. Another problem is the tendency to view activity as sequential when it is actually simultaneous. Although task work often entails sequential dependencies and can be reasonably decomposed by looking at individual capabilities, we cannot uniquely describe or design teamwork in this way. Teamwork is necessarily based on the interaction among the participants, whereas a simplifying notion of levels treats elements as cleanly separable.

Using Figure 3 as an example again, there seems to be a sequential ordering of the task elements. This might be appropriate for some tasks but not in general. Most teamwork occurs concurrently. Looking at the description of level 6 in the first column of Figure 3, it includes the phrase "informs the human in plenty of time to stop it." This implies the human is concurrently monitoring and assessing the computer's activity on some level. It would also suggest the need for a stop function, although none is included. The simplification here might explain the apparent oversight of including a stop behavioral element, and it is indicative of the problems faced when using a model with a solitary focus on levels of autonomy.

### *4.2.5 Problem 5: The Levels of Autonomy Concept Is Insufficient to Meet Future Challenges*

Many of the challenges facing designers are related to teamwork. An example is the proposed 10 challenges for making automation a "team player" (Klein, et al., 204). These challenges include directability, transparency, and predictability. These challenges deny the intrinsic validity of any levels of autonomy concept. Each of these challenges must be addressed not by making the machines more independent, but by making them more capable of supporting system interdependence.

Many supportive behaviors are what might be called soft system constraints and are not essential to task completion—that is, although the performer is, strictly speaking, self-sufficient, it can benefit from support. Joint activity is not exclusively about the hard constraints that enable or prevent the possibility of an activity, as the solid arrows in Figure 3 depict. Joint activity also includes soft interdependence, which includes optional commands, such as the ability to request the final status of the action (see the dashed arrow in Figure 3). Soft interdependence also includes helpful things that a participant might do to facilitate team performance. For example, team members can signal progress appraisals (Feltovich, et al., 2007) ("I'm running late"), warnings ("Watch your step"), helpful adjuncts ("Do you want me to pick up your prescription when I go by the drug store?"), and observations about relevant unexpected events ("It has started to rain").

Our observations suggest that good teams can be distinguished from great ones by how well they support requirements arising from soft interdependence. Although social science research on teamwork indicates it as an important factor in team performance (Salas, Cooke and Rosen, 2008), interdependence (particularly soft interdependence) has not received adequate attention in the research literature (Johnson, et al., 2012).

Consider the hypothetical level 6 in Figure 3. If we consider the interdependence in the activity, we can concoct a figure (Figure 4) patterned after the Sheridan-Verplank levels of automation. We have added some potential interdependence that might be appropriate for such an activity. We allow the sequential-work-flow assumption to persist only to maintain consistency in the discussion. The focus of Figure 4 is the diversity of interdependence among the activities.

Although we apply this process to a single level within the original Sheridan-Verplank list here, it can be applied to any of the levels with different results, based on the varying interdependence within the activity. If we move beyond the single decisive element portrayed by the Sheridan-Verplank list toward activity to support the future envisioned roles, the interdependence become much more complex and generating such a table becomes even more interesting. Such a construction calls out the ways in which changes to the level of autonomy affect interdependence and how the interdependence affects the total work system. Levels by themselves do not provide this information, which leads to the next problem.

| Things the human does | Things the human depends on the computer for | Things the computer depends on the human for | Things the computer does |
|---|---|---|---|
| Request options | Computer status and availability | | |
| | | Initiation by request<br>Environment status<br>Supervisor status | Get options |
| Request select action | A list of options<br>Factors effecting option selection<br>Ranking of options | | |
| | | Initiation by request<br>Preferences | Select action |
| Approve action | An action to approve<br>Reasoning behind decision<br>Confidence level | | |
| | | Approval<br>Constraints on activity | Start action |
| Request tell | Action status<br>Environment status<br>Progress appraisal | | |
| | | What to tell<br>When to tell<br>Who to tell<br>How to tell | Tell |

Figure 4 Example of an interdependence analysis based on the Figure 3 example with the addition of some potential interdependence. The solid arrows depict hard constraints, and the dashed arrow indicates soft interdependence. (Adapted from an earlier work Sheridan & Verplank, 1978)

### *4.2.6 Problem 6: Levels Provide Insufficient Guidance to the Designer*

Levels of autonomy do not provide principles or guidelines for designers as they build human-machine systems. Previous articles have discussed the challenge of bridging the gap from cognitive engineering products to software engineering (Hoofman, 2008). The levels of autonomy concept provides no assistance here. Parasuraman, Sheridan, and Wickens (2000) suggested using levels of autonomy in combination with human performance as an evaluative criterion for automation design. Although we agree that human-performance measures are important and useful, it is unclear what value the descriptive levels of autonomy provide other than as a labeling mechanism. They provide no assistance to the designer, whose only option is to build it and try it, then build something else and compare the results.

Interdependence, however, affords a great deal of predictive power. It can inform the designer of what is and is not needed, what is critical, and what is optional. Most importantly, it can indicate how changes in capabilities affect relationships.

This extends the human-centered approaches where designers typically ask, “How can we keep the human in the loop?” or “How do we reduce the burden on the human?” These types of questions lead designers to focus on usability issues. Understanding the interdependence in the human-machine system in the context of the anticipated activity can provide a wealth of guidance to a designer. In fact, we posit that it is through understanding the dynamic interdependence within the macrocognitive work that the system developer can answer such questions as “What should be automated?” and “How do we reduce the burden on the human?” More importantly, it has the potential to answer richer questions, such as “How will this change affect the work system?”

As an example, consider our level 6 in Figure 3. What is the impact of allowing the computer to move from the get options to select action functions without requiring the human request function? Making this change might enable a higher level of autonomy, but is it better? How does it affect the system?

Now look at Figure 4. Identifying the interdependence suggests several impacts. Not only does allowing the computer to select the action reduce the directability of the automation by eliminating the computer’s dependence on the human to initiate action selection, it also reduces transparency because the human no longer has access to the options. Both of these limit the work system’s ability to leverage the human’s ability to improve the overall work system’s effectiveness.

In summary, these six problems support the claim the concept of *levels of autonomy* is incomplete and insufficient as a model for designing complex human-machine teams.

## 4.3 Myths of Autonomy

We now explore some misconceptions surrounding the topic of “autonomous systems” and bust some "myths" of automation. As designers conceive and implement what are commonly (but mistakenly) called autonomous systems, they adhere to certain myths of autonomy that are not only damaging in their own right but also by their continued propagation, because they engender a host of other serious misconceptions and consequences.

### 4.3.1 *Myth #1: "Autonomy" is unidimensional.*

There is a myth that autonomy is some single thing and that everyone understands what it is. However, it is employed with different meanings and intentions (Bradshaw, Feltovich, et al., 2004). Beer, Fisk and Rogers (2012, pp 14-15) point out the "muddled use of the term." As an example, consider the self-sufficiency and self-directedness sense described previously. These two different senses affect the way autonomy is conceptualized and influence tacit claims about what "autonomous" machines can do. It should be evident that independence from outside control does not entail the self-sufficiency of an autonomous machine. Nor do a machine's autonomous capabilities guarantee that it will be allowed to operate in a self-directed manner. In fact, human-machine systems involve a dynamic balance of self-sufficiency and self-directedness. Capabilities machines have for autonomous action interact with the responsibility for outcomes and delegation of authority. Only people are held responsible for consequences (i.e., only people can act as problem holders) and only people decide on how authority is delegated to automata—see (Woods & Hollnagel, 2006, chapter 11)). It is more productive to think about autonomy in terms of multiple task-specific dimensions rather than in terms of a single, unidimensional scale (Bradshaw, Feltovich, et al., 2004).

### 4.3.2 *Myth #2. The conceptualization of "levels of autonomy" is a useful scientific grounding for the development of autonomous system roadmaps.*

A recent survey of Human-Robot Interaction observed that "perhaps the most strongly human-centered application of the concept of autonomy is in the notion of level of autonomy" (Goodrich & Schultz, 2007, p. 16). The survey highlights the proclivity of the concept of *levels of autonomy* in the HRI domain, but also suggests it is incomplete, recommending a different scale to address how the human and robot interact. A recent DSB report makes a more striking recommendation on the role of autonomy. It recommends that the "DoD should abandon the debate over definitions of levels of autonomy" (Defense Science Board, 2012, p. 2). The committee received inputs from multiple organizations on how some variation of definitions across levels of autonomy could guide new designs. The retired flag officers, technologists, and academics on the task force overwhelmingly and unanimously found the definitions irrelevant to the real problems, cases of success, and missed opportunities for effectively utilizing increases in autonomous capabilities for defense missions. This correlates with the first argument that the concept of *levels of autonomy* is incomplete and insufficient as a model for designing complex human-machine teams.

### *4.3.3 Myth #3: Autonomy is a widget.*

The Defense Science Board report points to the fallacy of "treating autonomy as a widget":

> The competing definitions for autonomy have led to confusion among developers and acquisition officers, as well as among operators and commanders. The attempt to define autonomy has resulted in a waste of both time and money spent debating and reconciling different terms and may be contributing to fears of unbounded autonomy. The definitions have been unsatisfactory because they typically try to express autonomy as a widget or discrete component, rather than a capability of the larger system enabled by the integration of human and machine abilities (p. 23).

In other words, we might say that autonomy is most usefully viewed as an emergent property of a system that functions capably in a given situation, and not as a description of the particular technology used to build it.

The myth of autonomy as a widget engenders the misunderstandings implicit in the next myth.

### *4.3.4 Myth #4. Autonomous systems are autonomous.*

Strictly speaking, the term "autonomous system" is a misnomer. Autonomy is not a property of a system, or a piece of technology, but rather is an idealistic characterization of the interactions among the machine, the task, and the situation. No entity—and, for that matter, no person—is capable enough to be able to perform competently in every task and situation. On the other hand, even the simplest machine can seem to function "autonomously" if the task and context are sufficiently constrained. A thermostat exercises an admirable degree of self-sufficiency and self-directedness with respect to the limited tasks it is designed to perform through the use of very simple form of automation (at least until it becomes miscalibated).

The Defense Science Board report wisely observes that "... there are no fully autonomous systems just as there are no fully autonomous soldiers, sailors, airmen, or Marines… Perhaps the most important message for commanders is that all machines are supervised by humans to some degree, and the best capabilities result from the coordination and collaboration of humans and machines." (p. 24).

What is the result of belief in this fourth myth? People in positions of responsibility and authority may over focus on autonomy-related problems and fixes while failing to understand that self-sufficiency is always relative to a situation. In fact, in most cases it is not only relative to a set of *predefined* tasks and goals, it is relative to a set of *fixed* tasks and goals. A software system might perform gloriously without supervision in circumstances within its competence envelope (itself a reflection of the designer's intent) but fail miserably when the context changes to some circumstance that pushes the larger work system beyond the edges of its competence envelope (Robert R Hoffman & Woods, 2011).

### *4.3.5 Myth #5: Once achieved, full autonomy obviates the need for human-machine collaboration.*

Autonomy research has been pursued in a technology-centric fashion, as if full autonomy—complete independence and self-sufficiency of each system—were the Holy Grail. The slogan in Figure 5 is an example of this sentiment. Wickens (Hancock et al., 2013) states that "a long-held conventional wisdom is that a greater degree of automation in human-in-the-loop systems produces both costs and benefits to performance." The ostensible reason for the quest is to reduce manning needs, since salaries are the largest fraction of the costs of sociotechnical systems. Of course, there are situations where the goal of minimizing human involvement with autonomous systems can be argued effectively—e.g., some jobs in industrial manufacturing. However, it should be noted that virtually all of the most challenging deployments of autonomous systems to date—e.g., military unmanned air vehicles, NASA rovers, unmanned underwater vehicles, and disaster inspection robots—have involved people in crucial roles. Such involvement has not been merely to make up for the current limitations on machine capabilities, but also because their jointly coordinated efforts with humans were—or should have been—intrinsically part of the mission planning and operations itself.


Figure 5 A (presumably) tongue-in-cheek Frisbee from Carnegie Mellon Robotics Institute. While humorous, it is the sentiment behind myth #5.

What is the result of belief in this myth? Researchers and their sponsors begin to assume that "all we need is more autonomy." This kind of simplistic thinking engenders the even more grandiose myth that human factors can be avoided in the design and deployment of machines. Beer et al. (2012) observe two dichotomous viewpoints: "(1) higher robot autonomy involves lower levels or less frequent HRI; and (2) higher robot autonomy requires higher levels or more sophisticated forms of HRI." Careful consideration will reveal that, in addition to more machine capabilities for taskwork, there is a need for the kinds of breakthroughs in human-machine teamwork that would enable autonomous systems not merely to do things *for* people, but also to work together *with* people and other systems. This capacity for teamwork, not merely the potential for expanded taskwork, is the inevitable next leap-forward required for more effective design and deployment of autonomous systems operating in a world full of people (Bradshaw, Carvalho, et al., 2012).

### *4.3.6 Myth #6: As machines acquire more autonomy, they will work as simple substitutes (or multipliers) of human capability.*

Function allocation is not a simple process of transferring responsibilities from one component to another. When a system designer automates a subtask, what he or she is really doing is performing a type of task distribution and, as such, has introduced novel elements of interdependence within the system (Johnson, Bradshaw, Feltovich, Jonker, et al., 2011). This is the lesson to be learned from studies of the "substitution myth" (Christoffersen & Woods, 2002) which conclude that reducing or expanding the role of automation in joint human-automation systems may change the nature of interdependent and mutually-adapted activities in complex ways. In order to effectively exploit the capabilities that automation provides (versus merely increasing automation), the taskwork—and the interdependent teamwork it induces among players in a given situation—must be understood and coordinated as a whole.

What is the result of belief in the myth of machines as simple multipliers of human ability? Because design approaches based on this myth do not adequately take into consideration the significant ways in which the introduction of autonomous capabilities can change the nature of the work itself, they lead to "clumsy automation." And trying to solve this problem by adding more poorly-designed autonomous capabilities, is, in effect, adding more clumsy automation onto clumsy automation, likely exacerbating the problem that the increased autonomy was intended to solve.

### 4.3.7 *Myth #7: "Full autonomy" is not only possible, but is always desirable.*

Ironically, even when technology succeeds in making tasks more efficient, the human workload is not reduced accordingly. David Woods and Erik Hollnagel (2006) summarized this phenomenon as the *law of stretched systems:* "every system is stretched to operate at its capacity; as soon as there is some improvement, for example in the form of new technology, it will be exploited to achieve a new intensity and tempo of activity" (2006, p. 18). All useful robotic endeavors, such as exploring mars or repairing the underwater oil rig during the gulf oil spill, are really human endeavors. As such, humans will always be involved. Striving for full autonomy is ignoring the contextual understanding and creativity people bring to a problem.

## 4.4 Conclusions

Though continuing research to make machines more active, adaptive, and functional is essential, the point of increasing such proficiencies is not merely to make the machines more *independent* during times when unsupervised activity is desirable or necessary (i.e., autonomy), but also to make them more capable of sophisticated *interdependent* activity with people and other machines when such is required (i.e., teamwork). Research in joint activity highlights the need for autonomous systems to support not only fluid orchestration of task handoffs among people and machines, but also combined participation on shared tasks requiring continuous and close interaction—i.e., coactivity (Johnson, Bradshaw, et al., 2012; Klein et al., 2004). Indeed, in situations of simultaneous human-agent collaboration on shared tasks, people and machines may be so tightly integrated in the performance of their work that interdependence is a continuous phenomenon and the very idea of task handoffs is incongruous.

The points raised focus on how to make effective use of the expanding power of machines. The myths we have discussed lead developers to introduce new machine capabilities in ways that predictably lead to unintended negative consequences. We need to discard the myths and focus on developing coordination and adaptive mechanisms that turn platform capabilities into new levels of mission effectiveness. In complex domains characterized by uncertainty, machines that are merely capable of performing independent work are not enough. Instead, we need machines that are also capable of working interdependently (Johnson, Bradshaw, et al., 2012).

For this to happen, we need an understanding of interdependence, system models based on interdependence and design methodologies that enable designers to support interdependence in their human-machine systems. This is precisely the goal of Coactive Design.

***Interdependence is and ought to be***
***as much the ideal of man as self-sufficiency.***

**– Mahatma Gandhi**

# 5 Coactive Design[8]

We propose Coactive Design as a new approach to address the increasingly sophisticated roles that people and robots play as the use of robots expands into new, complex domains. These roles involve humans and machines engaged in joint activity that can best be described as teamwork.

We coined the term *coactive* as a way of characterizing the activity. Besides implying more than one party is involved in the activity, the term “coactive” is meant to convey the type of involvement. Consider an example of playing the same sheet of music as a solo versus a duet. Although the music is the same, the processes involved are very different (Clark, 1996). The difference is that the process of a duet requires ways to support the interdependence among the players. This is a drastic shift for many autonomous robots, most of which were designed to do things as independently as possible.

The process of design is about developing something for an intended purpose. The term “coactive design” is about designing in a way that enables effective teamwork through support for interdependence. The goal of Coactive Design is to help designers identify interdependence relationships in a joint activity. This is so they can design, with a purpose, systems that support the relationships deemed appropriate. These supporting relationships thus enable designers to achieve the objectives of coordination, collaboration, and teamwork.

## 5.1 What it means to be coactive

The dictionary[9] gives three meanings to the word “coactive”: 1) Joint action, 2) An impelling or restraining force; a compulsion, 3) Ecology; any of the reciprocal actions or effects, such as symbiosis, that can occur in a community. These three meanings capture the essence of our approach and we translate these below to identify the three minimum requirements of a coactive system. Our contention is that for an agent to effectively engage in joint activity, it must at a minimum have:

[8] This chapter is adapted from (Johnson, Bradshaw, Feltovich, Jonker, et al., 2011)
[9] http://www.thefreedictionary.com/coactive (accessed on 16 February, 2014)

1) *Awareness of interdependence in joint activity*
2) *Consideration for interdependence in joint activity*
3) *Capability to support interdependence in joint activity*

Awareness corresponds to the first definition and is about knowledge of the joint action. We are not suggesting that all team members must be fully aware of the entire scope of the activity, but they must be aware of the aspects of the activity that are interdependent. Consideration for something implies that the thing could impel or restrain a decision. As with awareness, all team members do not need to be equally capable, but they do need to be capable of supporting their particular points of interdependence. The capacity to support interdependence requires reciprocal capabilities. We now address each requirement in more detail.

### *5.1.1 Awareness of Interdependence in Joint Activity*

In human-machine systems like today's flight automation systems, there is a shared responsibility between the humans and machines, yet the automation is completely unaware of the human participants in the activity. Joint activity implies mutual engagement in a process extended in space and time (Klein et al., 2005; Sierhuis, 2007). Previous work in robotic systems has focused largely on assigning or allocating tasks to machines that may know little about the overall goal of the activity or about other tasks on which its tasks may be interdependent. This approach underpins supervisory control which focuses on "what to automate and to what extent (Parasuraman et al., 2000)" and results in approaches that view the role of HRI as identifying "appropriate trade-offs in allocating tasks to either a human or a robot (Beer et al., 2012)." A recent survey of supervisory control frameworks states that a significant fault with these frameworks is that they are "based on a hierarchical task decomposition to describe the delegation relationship (Miller, 2012, p. 186)" and focus "only the act of delegating, not the context in which that act occurs (Miller, 2012, p. 186)." Similarly, Cummings et al. note that current analysis methods are limited by "focusing on the needs of the individual team members, often ignoring the collective decision making and coordination that is actually required (Cummings, da Silva, & Scott, 2007, p. 8)." However, the increasing sophistication of human-machine systems depends on a mature understanding of the requirements of interdependence between team members in joint activity. As such, humans should be "integral system components rather than system users. (Adams et al., 2009, p. 20)" We are no longer dealing with individual *autonomous* actions but with group *participatory* actions (Clark, 1996).

### *5.1.2 Consideration for Interdependence in Joint Activity*

Awareness of interdependence is only helpful if that awareness has the potential to alter decisions. This means that requirements for interdependence must be taken into account in the design of autonomous capabilities. As an example, consider playing the same sheet of music as a musical solo versus a duet. Although the music is the same, the processes involved are very different. As Clark (1996) states, "a person's processes may be very different in individual and joint actions even when they appear identical." This type of consideration can be found in some dialogue systems (e.g. Cantrell, Scheutz, Schermerhorn, & Wu, 2010) where the domain forces interdependence to the forefront. It can also be seen in human-centered design approaches that try to account for the human role and limitations by fitting systems to users (e.g. Adams et al., 2009; Cooper, 2007; Goodrich, 2004) and systems that aim to supplement limitations in robotic systems via human support (e.g. Fong, 2001; Michaud et al., 2010). However, the issue of consideration has been noted by several in the HRI community including Macbeth et al. who state "typically algorithm designers generate the optimization algorithm first, and then the interface designers are left to support the operator with often incomplete information because the interface requirements of the human were not considered at the time of algorithm generation (Macbeth, Cummings, Bertuccelli, & Surana, 2012, p. 2348)." Designing systems, both the autonomy and interface, that address the requirements for interdependence is a drastic shift for autonomous robots, most of which were designed to do things as independently as possible.

In addition to the processes involved being different, joint activity is inherently more constraining than independent activity. Joint activity may require participating parties to assume collective obligations (Diggelen, Bradshaw, Johnson, Uszok, & Feltovich, 2009) that come into play even when they are not currently "assigned" to an ongoing task. These obligations may require the performance of certain duties that facilitate good teamwork or they may limit our individual actions for the good of the whole. For example, we may be compelled to provide help in certain situations, while at the same time being prevented from hogging more than our share of limited resources. In joint activity, individual participants share an obligation to coordinate; sacrificing to a degree their individual autonomy in the service of progress toward group goals. These obligations should not be viewed as only a burden. While it is true they usually have a cost, they also provide an opportunity. Ensley (1999, p. 490) notes that "implementation strategies that provide assistance with the manual workload associated with a task while still keeping the operator involved in current operations appears to be optimal."

### *5.1.3 Capability to Support Interdependence in Joint Activity*

While consideration is about the deliberative or cognitive processes, there is also an essential functional requirement, referred to as teamwork infrastructure in Chapter 1.4. We have described self-sufficiency as the capability to take care of one's self. Here we are talking about the capability to support interdependence.

This means the capability to assist another or be assisted by another. The coactive nature of joint activity means that there is a reciprocal requirement in order for interdependence to be supported, or to put it another way, there is the need for complementary capabilities of those engaged in a participatory action. For example, if I need to know your status, you must be able to provide status updates. If you can help me make navigation decisions, my navigation algorithm must allow for outside guidance. Simply stated, one can only give if the others can take and vice versa. Similar observations have been made by others. Beer et al. states that "the proper match between the level of robot autonomy and the method of control is essential. (2012, p. 61)" Crandall et al. pragmatically note "Because both robot autonomy and the interface dictate the human–robot interactions, they should be designed together. (J. W. Crandall, Goodrich, Olsen, & Nielsen, 2005, p. 438)" The abilities required for good teamwork require complementary abilities from the participating team members.

## 5.2 The Fundamental Principle of Coactive Design

The fundamental principle of Coactive Design is that interdependence must shape autonomy. Certainly joint activity of any consequence requires a measure of autonomy (both self-sufficiency and self-directedness) of its participants. Without a minimum level of autonomy, an agent will simply be a burden on a team, as noted by Stubbs (2007). However, it can be shown that in some situations simply adding more autonomy can hinder rather than help team performance. The means by which that agent realizes the necessary capabilities of self-sufficiency and self-directedness must be guided by an understanding of the interdependence between team members in the types of joint activity in which it will be involved. This understanding of interdependence can be used to shape the design and implementation of the agent's autonomous capabilities, thus enabling appropriate interaction with people and other agents.

In contrast to autonomous systems designed to take humans out of the loop, we are specifically designing systems to address requirements that allow close and continuous interaction with people. This is important because of trends observable in technology development. Consider the history of research and development in unmanned aerial vehicles (UAVs), depicted in Figure 6. The first goal in its development was a standard engineering challenge to make the UAV self-sufficient for some tasks (e.g., stable flight, waypoint following). As the capabilities and robustness increased, the focus shifted to the problem of self-directedness (e.g., what am I willing to let the UAV do autonomously). The future directions of UAVs indicate a another shift, as discussed in the Unmanned Systems Roadmap (Department of Defense, 2007) which states that unmanned systems "will quickly evolve to the point where various classes of unmanned systems operate together in a cooperative and collaborative manner…" This suggests a need to focus on interdependence (e.g., how can I get multiple UAVs to work effectively as a team with their operators?). This pattern of development is a natural maturation process that applies to any form of sophisticated automation. As we try to design more sophisticated systems, we move along a maturity continuum[10] from dependence to independence to interdependence (Figure 6). The process is a continuum because a small level of independence of machines through autonomy is a prerequisite for interdependence. However, independence is not the supreme achievement in human-human interaction (Covey, 1989), nor should it be in human-machine systems. This can be seen historically in the developmental trends of robotic systems. Future robots are intended for more sophisticated tasks that have a potentially high degree of interdependence. While awareness of interdependence may not critical to the initial stages of system development, it becomes an essential factor in the realization of a system's full potential.

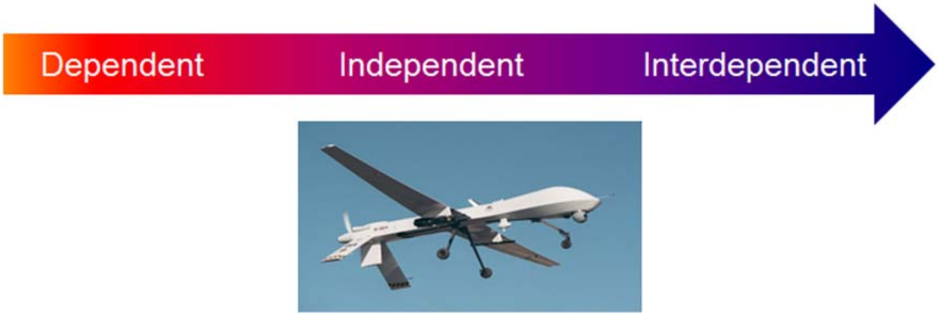



Figure 6 The natural maturation process of technology — from dependence to independence to interdependence. While awareness of interdependence may not critical to the initial stages of system development, it becomes an essential factor in the realization of a system's full potential.

## 5.3 A New Perspective

The perspective afforded by Coactive Design helps understand the challenges of current approaches and suggests new ways to address those challenges. First, consider the main questions plaguing roboticists. These questions are shown on the left hand side of Table 3 and include questions like "what is the robot doing?" and "what is it going to do next?" The underlying issues for each of these questions are listed in the middle column of Table 3 and include things like transparency and predictability. These issues are issues of supporting interdependence, not issues of

[10] This is adapted from Stephen Covey's maturity continuum for personal effectiveness. We have extended this to an observed pattern we have noticed in the development of technology.

autonomy. To highlight the complementary nature of interdependence, we also included the robot needs or challenges. These are shown in the right most column of Table 3 and include things like understanding the intent of the human and knowing if the human can provide assistance. These needs correlate with the human needs and are also addressed through support for interdependence.

Table 3 Common robot questions (left). The underlying issues (middle) are all about interdependence, not autonomy. The robot has needs (right) that correlate with the human questions and are also addressed through interdependence.

| Human Needs | Issues | Robot Needs |
|---|---|---|
| What is the robot doing? | Mutual Transparency | What is the intent of the human? |
| Why did the robot do that? | Mutual Explainability | What is the task context? |
| What is the robot going to do next? | Mutual Predictability | What does the human need from me? |
| How can we get the robot to do what we need? | Mutual Directability | Can the human provide help? |
| Does use of autonomy add value? | Mutual Cost Benefit Management | Will my actions provide value to the human? |

Another way to visualize how the Coactive Design perspective aligns with current perspectives is to understand the challenges faced by autonomy-centered approaches. The two senses of autonomy are shown as a graph in Figure 7. Since the capability to perform a task and the authority to perform a task are orthogonal concepts, we separate these two dimensions onto separate axes. Together these two axes represent an autonomy-centered plane of robotic capabilities. The *self-sufficiency* axis represents the degree to which a robot can perform a task by itself. "Low" indicates that the robot is not capable of performing the task without significant help. "High" indicates that the robot can perform the task reliably without assistance. The *self-directedness* axis is about freedom from outside control. Though a robot may be sufficiently competent to perform a range of actions, it may be constrained from doing so by a variety of social and environmental factors. "Low" indicates that, although possibly capable of performing the task, the robot is not permitted to do so. "High" indicates the robot has the authority over its own actions, though it does not necessarily imply sufficient competence.

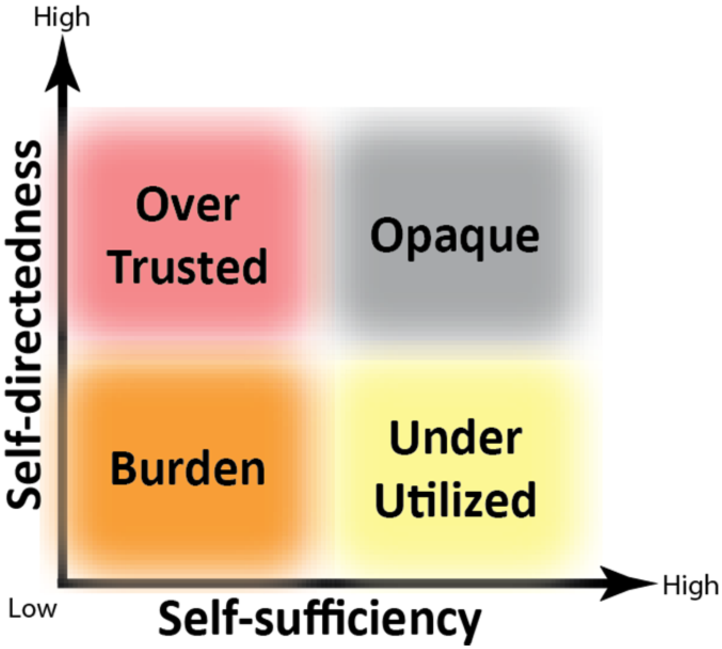



Figure 7 Common system issues mapped against an autonomy-centered plane.

Direct teleoperation, in which both self-sufficiency and self-directedness are absent, corresponds to the region labeled *Burden*. By burden, we are referring to the workload imposed on the operator by the high robot attention demand (Olsen & Goodrich, 2003) and the human performance issues associated with remote operation (Chen, Haas, & Barnes, 2007). These combine to consume the operator's attention with attending to the robot and leave little remaining attention to be directed toward the mission (Burke, Murphy, Coovert, & Riddle, 2004). Increasing the self-directedness without a corresponding level of self-sufficiency will result in a system that is *over-trusted*, as shown in the upper left of the figure. Over-trust is also a generalization that is meant to include any time automation is relied upon and permitted to exceed its own capabilities. A significant amount of research has focused on understanding the perils of introducing automation into the aviation domain (Kaber, Riley, Tan, & Endsley, 2001; McCarley & Wickens, 2005; Norman, 1990; Sarter & Woods, 1995; Woods & Sarter, 1997) as well as many other complex domains (Bainbridge, 1983; Blackhurst, Gresham, & Stone, 2011; Klein, Woods, Bradshaw, Hoffman, & Feltovich, 2004; Perrow, 1984; Woods & Branlat, 2010). When autonomous capabilities are seen as insufficient, particularly in situations where the consequences of robot error may be disastrous, it is common for self-directedness to be limited. When the system self-directedness is reduced significantly below the potential of its capabilities the result is an *underutilized* system, as shown in the lower right corner of the figure. An example of this would be the first generations of Mars rovers which, due to the high cost of failure, were not trusted with autonomous action, but rather were subject to the decisions of a sizable team of NASA engineers. Here is the key point, however:

> *Even when self-directedness and self-sufficiency are reliable, matched appropriately to each other, and sufficient for the performance of the robot's individual tasks, human-robot teams engaged in consequential joint activity frequently encounter the potentially debilitating problem of opacity, meaning the inability for team members to maintain sufficient awareness of the state and actions of others to maintain effective team performance.*

The problem of *opacity* in robotics was highlighted recently by Stubbs (2007) but had been previously identified as a general challenge more than two decades ago by Norman (1990). Norman cites numerous examples of opacity, most of which come from aviation where silent (opaque) automation has led to major accidents. This opacity often leads to what Woods (1997) calls "automation surprises" that may result in catastrophe. An example is an autopilot that silently compensates for ice build-up on the airplane wings, while pilots remain unaware. Then, when the limits of control authority are reached and it can no longer compensate for extreme conditions, the automation simply turns off, forcing the pilots to try to recover from a very dangerous situation. It is important to recognize that the challenges go far beyond simply not being able to see needed information. They can also involve predictability, directability or other challenges that must be addressed in order to turn autonomous systems into team players (Klein et al., 2004).

So how does the coactive design perspective change the way we see the design problem? So far, we have depicted the two senses of autonomy on two orthogonal axes representing an autonomy-centered plane of agent capabilities. Coactive Design adds a third orthogonal dimension of agent capability: support for interdependence (Figure 8).

The *support for interdependence* axis characterizes an agent in terms of its capability to depend on others or be depended on by others in any of the dimensions of autonomy. This axis is specifically about the capability to be interdependent, *not* the need or requirement to *be* dependent which are captured by the other axes. Although we are showing a single set of axes for simplicity, there are many dimensions to autonomy (Bradshaw et al., 2004).

As we look at the challenges faced by current autonomous systems from a Coactive Design perspective, we see not only the constraints imposed by interdependence in the system, but also as a tremendous opportunity. Instead of considering the activity an independent one we can think about it as a participatory one (Clark, 1996). Both the human and the machine are typically engaged in the same activity. There may be domains where we would like a robot to go on its mission and simply return with a result, but most domains are not like this. We need the agent to have some self-sufficiency and self-directedness, but we remain interdependent as the participatory task unfolds. Supporting this need provides an opportunity to address some of the current challenges. Figure 8 lists just a few such opportunities. For example, over-trusted robots can be supplemented with human assistance and opaque systems can provide feedback and transparency. In fact, many of the ten challenges (Klein et al., 2004) of automation, such as predictability and directability apply to this new dimension, as do all of the issues in Table 3.

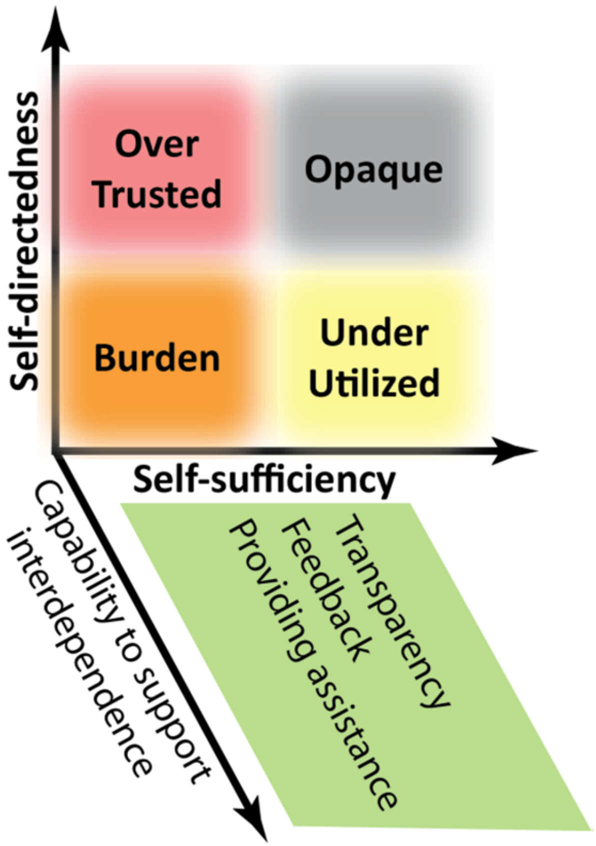



Figure 8 Support for interdependence as an orthogonal dimension to autonomy and some opportunities this dimension offers

We can now map examples of prior work in autonomy onto this space (Table 4). In the Related Work section, we described how previous work was focused on self-sufficiency and self-directedness. Coactive Design presents the unique perspective of the *support for interdependence* dimension which is captured in the two rightmost columns of Table 4: the ability to depend on others and the ability to be depended on by others. The most important innovation of the Collaborative Control (Fong, 2001) approach was in accommodating a role for the human in providing assistance to the robot at the perceptual and cognitive levels. In other words, the robot had the ability to depend on the human for assistance in perception. The key insight of Collaborative Control was that tasks may sometimes be done more effectively if performed jointly. Coactive Design extends this perspective by providing a complement of this type of interdependence, accommodating the possibility of machines assisting people.

Table 4 Scope of concerns addressed by different approaches.

| Approach | Autonomy-Centered | | Teamwork-Centered (Support for Interdependence) | |
|---|---|---|---|---|
| | Self-sufficiency | Self-directedness | Ability to depend on others | Ability to be depended on |
| Teleoperation | | | | |
| Functional Allocation | ✓ | | | |
| Supervisory Control | ✓ | | | |
| Adjustable Autonomy | ✓ | ✓ | | |
| Sliding Autonomy | ✓ | ✓ | | |
| Adaptive Autonomy | ✓ | ✓ | | |
| Flexible Autonomy | ✓ | ✓ | | |
| Mixed Initiative Interaction | ✓ | ✓ | | |
| Shared Control | ✓ | ✓ | | |
| Collaborative Control | ✓ | ✓ | ✓ | |
| Coactive Design | ✓ | ✓ | ✓ | ✓ |

***Interdependence is a higher value than independence.***

**– Stephen R. Covey**

# 6 Interdependence[11]

Coactive Design takes *interdependence* as the central organizing principle among people and agents working together in joint activity. Our sense of joint activity parallels that of Clark (1996), who has described what happens in situations when what one party does depends on what another party does (and vice-versa) over a sustained sequence of actions. In such joint activity, we say that team members are "interdependent" (Feltovich et al., 2007). In his seminal book, James D. Thompson (1967) recognized the importance of interdependence in organizational design. Similarly, we feel that understanding interdependence is critical to the design of human-machine systems. The reason it is so important in all of these domains is because *interdependence is the basis for understanding complex systems*.

Thompson (1967) also noted that there was a lack of understanding about interdependence, which is still true today. Much work being done in this area focuses on teams of people (e.g. Cohen & Levesque, 1991; Fiore, 2008; Saavedra, Earley, & Van Dyne, 1993; Salas, Bowers, & Edens, 2001; Salas et al., 2008; Salas & Fiore, 2004) though there has been an effort to bridge these conceptual understandings to human-robot, human-agent and agent-agent systems (e.g. Breazeal et al., 2004; Cuevas, Fiore, Caldwell, & Strater, 2007; Jonker, Van Riemsdijk, & Vermeulen, 2011; Sierhuis et al., 2003; Sycara & Sukthankar, 2006; Sycara, 2002; Tambe, 1997). Understanding the nature of the interdependencies among groups of humans and machines provides insight into the kinds of coordination that will be required. Indeed, we assert that coordination mechanisms in skilled teams arise largely because of such interdependencies (Johnson et al., 2011). For this reason, understanding interdependence is an important requirement in designing machines that will be required to work as part of human-machine systems engaged in teamwork.

This thesis puts forth the concept that managing interdependent activities is the key design element for developing collaborative systems. This is because managing interdependencies is the mechanism by which we achieve the higher level concepts of coordination, collaboration and teamwork. As the mechanism, it provides a way to ground the conceptual into specific implementations.

[11] This chapter is adapted from: (Johnson et al., 2014)

## 6.1 What It Means To Be Interdependent

Many misunderstandings in science come from assigning different meanings to terms. With respect to the topic at hand, the terms “interdependence” and “dependence” are used in a range of ways with varied meaning—sometimes even as synonyms as in Malone and Crowston’s (1994) work[12]. In order to avoid confusion, we will clarify our interpretation of these terms by providing definitions. This is done with the understanding that others may hold different views on the meaning of the terms but can interpret our results through the lens we have provided.

Interdependence is often simply equated to mutual dependence[13]. However, this definition of the concept is too simplistic to capture the kinds of nuances we have observed in interdependence relationships among humans and machines engaged in joint activity. Thompson’s (1967) famous Organizational Theory work on interdependence states:

> It appears that if we wish to understand organization structure, we must consider what is meant by interdependence and by coordination, and we must consider the various types of these. (p. 54)

Thompson’s work provided insight into coordination mechanisms and outlined three types of interdependence, but did not provide a definition of interdependence itself. From the perspective of social psychology, Thibaut and Kelly’s (1959) Theory of Interdependence describe interdependence in the following way:

> In any dyad both members are dependent upon the relationship to some degree, so we speak of their being interdependent. This means that each one has some power over the other which places limits on the extent to which each may with impunity exercise his power over his colleague. The pattern of interdependency which characterizes a relationship also affects the kinds of process agreements the pair must achieve if their relationship is to be maximally satisfactory. (p.124)

This description suggests that interdependence is about *relationships*, which we feel is an important insight because it breaks the “black-box” mentality commonly found in robotics. Autonomy-centered perspectives can lead to a design process that is focused on building “black-boxes” that provide an algorithm to perform an action. By focusing on interdependence relationships it becomes clear that “both robot autonomy and the interface dictate the human–robot interactions, they should be designed together (J. W. Crandall et al., 2005)” which is a sentiment being

---

[12] Malone uses dependencies instead of interdependencies, but then states, “if there is no interdependence, there is nothing to coordinate.” This implies that he considers the two terms to be synonymous.

[13] http://dictionary.reference.com/browse/interdependent?s=t (accessed 17 November 2013).

echoed by an increasing number of researchers (e.g. Adams et al., 2009; Cooper, 2007; Johnson, Bradshaw, Feltovich, Jonker, et al., 2011; Macbeth et al., 2012). Thibaut and Kelly's description, however, is limited to one type of interdependence, specifically how an individual's behavior affects the outcomes of contingent relationships. Descriptions of other types of interdependence are provided by Malone and Crowston's (1994) interdisciplinary study of coordination:

> Coordination means "managing dependencies between activities." Therefore, since activities must, in some sense, be performed by "actors," the definition implies that all instances of coordination include actors performing activities that are interdependent. (p. 101)

From this, we glean that the purpose of these relationships is to *manage dependencies*, in this case interdependence among activities. Dependencies among agents have been an important theme in Distributed Artificial Intelligence research. For example, Jennings (1996) states:

> The nature of the inter-agent dependencies is the critical determinant of the type of coordination which will take place. (p. 5)

This statement emphasizes the importance of the concept of interdependence, but provides no definition of the term. More recently, these concepts have begun to make their way into the HRI domain, as evidenced by Murphy and Burke's (2008) comment:

> An examination of team processes is useful because it identifies the dependencies between the agents in the system and how the agents are coordinated. This is key to designing systems that facilitate coordination. (p. 2)

This comment highlights the importance of identifying dependencies between agents for facilitating coordination. Again, however, the concept remains undefined.

Our definition of interdependence builds on the idea that interdependence is about relationships. It includes the purpose of these relationships which is to manage dependencies in joint activity. We emphasize that some dependencies are "hard" (absolutely necessary for carrying out the joint activity) while others are "soft" (defining possible opportunities for improving joint activity). In light of these considerations, we define interdependence in the context of joint activity as follows:

*"Interdependence" describes the set of complementary relationships that two or more parties rely on to manage required (hard) or opportunistic (soft) dependencies in joint activity.*

In the next section, we decompose our definition of interdependence, expounding the key notions in this definition. We begin with the term "dependence."

### *6.1.1 Dependence Is About Capacity*

In order to define dependence, we introduce the notion of *capacity*. Consider a robot that exists in some world environment and can sense and act on the world. A robot may require various things such as knowledge, skills, abilities, or resources to perform an activity. We define capacity as an encompassing term:

*Capacity is the total set of inherent things (e.g., knowledge, skills, abilities, and resources) that an entity requires to competently perform an activity individually.*

All aspects of capacity are determined by the interaction between a robot and its environment. It concerns the inherent capabilities of an entity and can be associated with the descriptive dimension of autonomy (Bradshaw et al., 2004). Consider the example of a robot that can deliver a soda. The most prominent aspects of capacity are the skills and knowledge to perform a task. This task requires skills such as planning a path to the refrigerator, moving to the refrigerator and avoiding obstacles along the way, opening the refrigerator, and picking up the soda can. It also requires knowledge of the refrigerator containing soda, where it is located, where the robot is relative to the refrigerator, and how to identify the desired can of soda. Capacity also includes accounting for resources such as energy and time. A robot may have the ability to get a soda, but its remaining battery life may not be sufficient. Similarly, it may be able to get a soda, but its maximum speed might hinder accomplishing this in under 30 seconds. Based on this view of capacity, we define dependence and its complement independence:

*Dependence exists when an entity lacks a required capacity to competently perform an activity in a given context.*

*Independence exists when an entity possesses the required capacity to competently perform an activity in a given context.*

Both dependence and independence can be interpreted using the concept of a control loop. Much of robotics is built on the concept of a control loop. Even Sheridan's (2011) latest work uses the control loop as the basis for comparing several common concepts and model frameworks of human–machine interaction. In some sense, being independent can be thought of as an agent having the capacity to close the behavioral sense-act loop, visualized as a complete green oval in Figure 9. In the dependence example, the sense-act loop was incomplete due to some lack of capacity, as shown by the red oval interrupted by the lacking capacity in Figure 9.

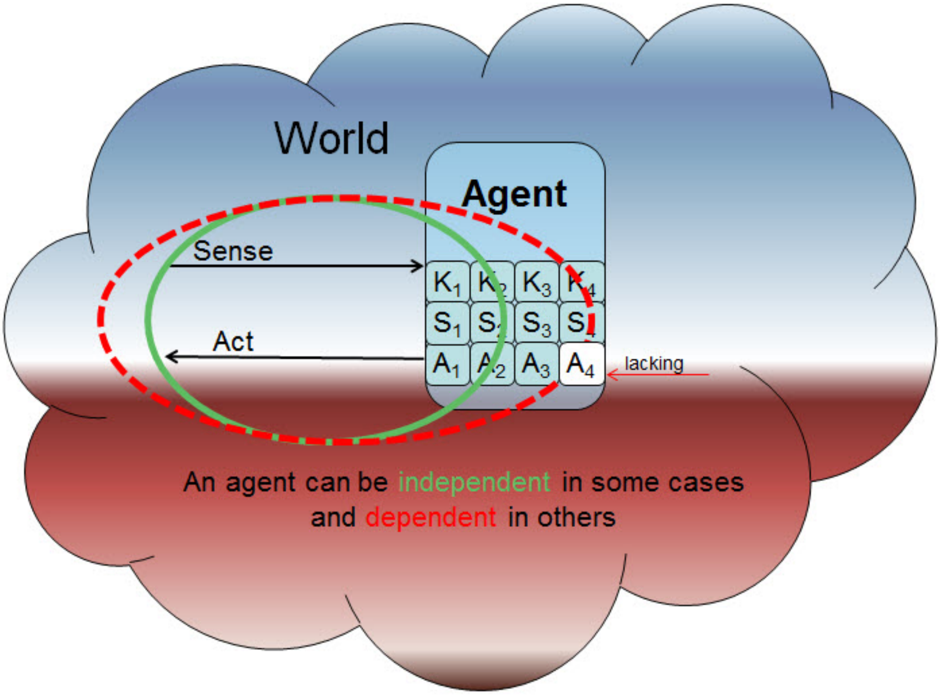



Figure 9 Dependence and Independence interpreted as control loops

Indigenous capabilities of an agent are multi-faceted and not a unitary component, indicated by the multiple boxes inside of the agent in Figure 9. An agent's capacity is comprised of multiple types of knowledge, skills and abilities and these determine what the agent is capable of accomplishing independently. This means an agent can be independent in some cases and dependent in others, as shown in Figure 9. It can be useful to think of the multiple facets of dependence metaphorically as a jigsaw puzzle piece. The structure indicates areas of independence and the voids in the piece indicate areas of dependence, as shown in Figure 10.

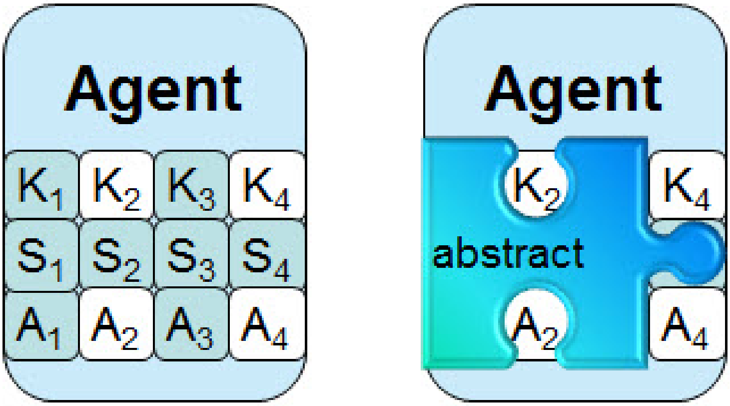



Figure 10 Puzzle piece metaphor for capacity

Besides being multi-faceted, capacity is also complicated because it is context dependent. An agent's capacity is typically described abstractly (e.g. the robot can pick up blocks). This is represented by the base capacity of the agent, shown as a blue puzzle piece in Figure 11. As capacity is applied in a concrete instance of an activity, context can inhibit or enhance the abstract capacity (e.g. I cannot lift blocks weighing over 100lbs.). This is represented by the yellow activity section of Figure 11 and the associated yellow context overlay on the blue puzzle piece indicating areas of the abstract capacity which are still viable in the context of the specific activity. Context can come from a variety of sources besides the activity, which we label the situation in Figure 11. This includes the environment, the weather, the history, and anything else that might affect abstract capacity (e.g. I can lift blocks weighing over 100lbs if they are under water). The surrounding green context in Figure 11 has an associated green puzzle piece overlay indicating how context inhibits or enhances capacity.

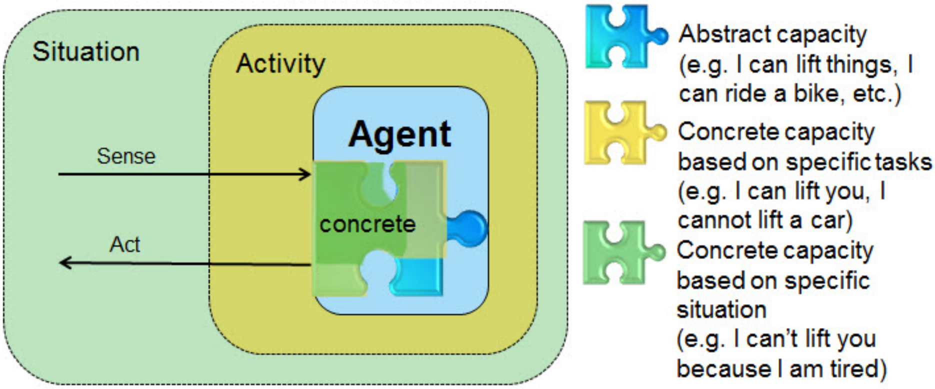



Figure 11 Capacity and the importance of context

In summary, dependence with respect to capacity is important to robot designers, as the goal of making a robot more autonomous requires designers to develop the necessary capacity for independence. However, designing for teamwork requires designing for interdependence, not just independence.

### *6.1.2 Interdependence Is About Relationships*

In our treatment, the concept of dependence with respect to capacity does not include other agents or their abilities, nor does it include interactions with other agents. These interactions or relationships are commonly described as one being "dependent *on another*." This sense of dependence can be associated with the prescriptive dimension of autonomy (Bradshaw et al., 2004). Examples include synchronized movements, delegations, and authority structures to, for example, permit or prohibit various actions. These all play the role of external regulatory systems, by which we mean any set of devices that serves to constrain or promote behavior in some direction (Feltovich et al., 2007). While it is perfectly fine linguistically and otherwise to use dependence to describe both senses of dependence, we will refer to this second sense of dependence with respect to regulatory relationships as interdependence.

Both senses of dependence provide reasons for establishing a relationship. However, a fully defined interdependence relationship includes both the reason for it (i.e., what is it trying to address) and the remedy (i.e., how is it going to be addressed). The reason for the relationship can include a capacity limitation or a regulatory relationship. The remedy is provided by creating mechanisms that support an interdependent relationship and *these mechanisms are the creative medium of the designer*. It is these mechanisms that compose the teamwork infrastructure (Figure 2) by which we can add not only capability, but also flexibility and resilience to a system.

The main issue that complicates understanding the concepts of dependence and interdependence is the cascading nature of the two concepts. Dependence with respect to capacity is managed by establishing supporting interdependent relationships. Conversely, establishing an interdependent relationship can impose new dependencies, as shown in Figure 2. These can cascade as the designer makes choices, so that a dependency inspires an interdependent relationship, which creates other dependencies, requiring additional interdependent relationships and so on.

As a simple example of cascading requirements, consider a blind person and a guide dog. There is an initial dependence with respect to capacity: The person lacks the ability to see. This is the reason for the dependence. The remedy is to establish a relationship with a guide dog to provide navigation support. This relationship, in turn, creates a new dependency based on the need to control where the dog guides the person. The first dependency, needing navigation support, is based on capacity (i.e., the person not being able to see) and was inherent in the problem. The second dependency, needing control guidance, is based on obligations incurred by establishing an interdependent relationship (i.e., the dog will guide the person). Notice that the second dependency did not exist until we established the interdependent relationship and it would cease to exist if the dog were no longer needed for guidance. The second dependency is a product of the remedy.

### *6.1.3 Interdependence Relationships Must Be Complementary*

To be a complement means to complete something[14]. There are two different ways interdependence can relate to being complementary; the connections and the capabilities.

In order to support an interdependent relationship the connections, as described in Chapter 1.4, need to be designed to work together. We refer to this as the endpoints being complementary. Using a train coupling shown in Figure 12 as a metaphor, each car must have a coupling that is designed to fit into the other properly, perform the function of holding the two cars together and both must be able to support the weight. In other words, the complete coupling is composed of the two endpoints. Sometimes designing a connection that is complementary is trivial. From the example in Chapter 1.4, providing a start button that allows the human task the robot and an algorithm to trigger when that button is pushed is indeed trivial. Other times, ensuring a complementary relationship is not so clear. How do you convey the amount of uncertainty a control algorithm has and when is

[14] Definition from http://www.thefreedictionary.com/complement (accessed on 05JULY2014).

it relevant to the human? How do you provide an appropriate appraisal of current robot status if the robot has no sensors to measure status? How do you provide the intent of the human to an algorithm? These are more challenging interdependencies to support. Regardless of the complexity, the connections of interdependent relationships must be complementary.

For capabilities, the requirement is not as stringent. There are cases where capabilities of the human and robot overlap completely and either can do the task without requiring the other. However, some cases rely on the paring of capabilities and the relationship must be complementary capabilities. As a simple example, imagine a train engine pulling a cargo car (as depicted in Figure 12). This situation would commonly be described as the cargo car being dependent on the engine to move. Based on our definitions, the car is indeed dependent (i.e., lacking capacity) and the two parties are interdependent (i.e., there is a relationship). This relationship is not as simple as the engine providing power for the car. The engine relies on the car to provide the cargo capacity. Note that this example relies on there being a joint activity of the train (consisting of cargo car and engine) to move some cargo from A to B. If there would not be such a joint activity, the engine could just move without the car. Therefore, joint activity is the assumption under which interdependence is defined, and as a result, interdependence must be complementary in this case.


Figure 12 Two train analogies describing the complementary nature of interdependence. The coupling is a complementary connection. The car and engine are complementary capabilities.

### *6.1.4 Interdependence Concerns Both Required (Hard) and Opportunistic (Soft) Relationships*

Much of the robotics work today is about required (i.e., hard) interdependence relationships that stem from lack of capacity, e.g., the train analogy depicted on the left side of Figure 13. However, it is our view that to achieve true teamwork, interdependence should also include opportunistic (i.e., soft) interdependence relationships. Soft interdependence does not stem from a lack of capacity. It arises from recognizing opportunities to be more effective, more efficient, or more robust by working jointly, as depicted on the right side of Figure 13.

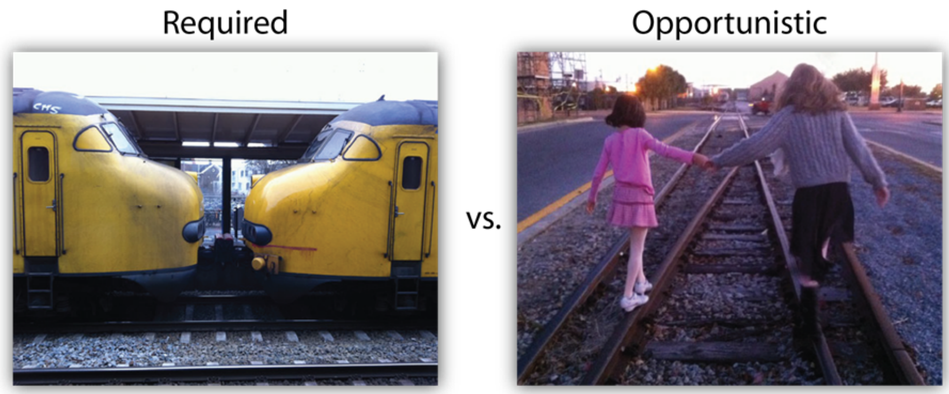



Figure 13 Hard (required) versus soft (opportunistic) interdependence relationships.

Soft interdependence is optional and opportunistic rather than strictly required. It includes a wide range of helpful things that a participant may do to facilitate team performance. Examples include progress appraisals ("I'm running late"), warnings ("Watch your step"), helpful adjuncts ("Do you want me to pick up your prescription when I go by the drug store?"), and observations about relevant unexpected events ("It has started to rain"). The importance of these different types of monitoring and feedback are well documented in human team literature (Guzzo & Salas, 1995; Larson & LaFasto, 1989; McIntyre & Salas, 1995; Smith-Jentsch, Baker, Salas, & Cannon-Bowers, 2001). All of these examples and many others reinforce the need to consider all internal cognitive processes of the parties involved, not just the interactive ones supporting hard constraints such as a lack of capacity. "It is not sufficient that members be technical experts – they must also be experts in the social interactions that lead to adaptive coordination action (i.e. teamwork)(Salas et al., 2006)."

Many aspects of teamwork are best described as soft interdependencies, which suggest the importance of designing mechanisms to support them. In studying expert performance in human teams, Salas et al. observed that "Expert teams create mechanisms for cooperation and coordination (Salas et al., 2006)." This is consistent with our observations to date which suggest that good teams can often be distinguished from great ones by how well they manage soft interdependencies.

### *6.1.5 Interdependence Recasts Context and Determines How Common Ground Can Be Reached*

Context matters and interdependent relationships in support of joint activity recast context. Instead of single agents in the context of individual activities, in teamwork all parties involved will be in the context of the same joint activity. In addition to each agent having its own situation, there is now a need to be aware of the situation enveloping all parties, as depicted in Figure 15. Consider an example of playing the same sheet of music as a solo versus a duet. Clark (1996) observes that "a person's processes may be very different in individual and joint actions, even when they appear identical." The difference is that the process of a duet requires ways to support the interdependence among the players, hence the interdependent relationship of playing as a duet recasts the context in which the sheet of music is played.

With new context comes a new definition of what is pertinent. Clark states that "all collective actions are built on common ground (Clark & Brennan, 1991)." Common Ground refers to the pertinent mutual knowledge, mutual beliefs and mutual assumptions that support interdependent actions in some joint activity (Clark & Brennan, 1991; Klein et al., 2004). The challenge with common ground is that it requires a grounding process (Brennan, 1998; Klein et al., 2005) to establish and maintain it. People develop the capabilities to support this process through social engagements throughout their life, but robots require support mechanisms to be designed into them. A field study by Stubbs et al. state that "users collaborating with the remote robot showed differences in how the users reached common ground with the robot in terms of an accurate, shared understanding of the robot's context, planning, and actions (Stubbs et al., 2007)." This is consistent with the three key interdependence relationships (observability, predictability and directability discussed in the next chapter) we have identified as key to reach common ground.

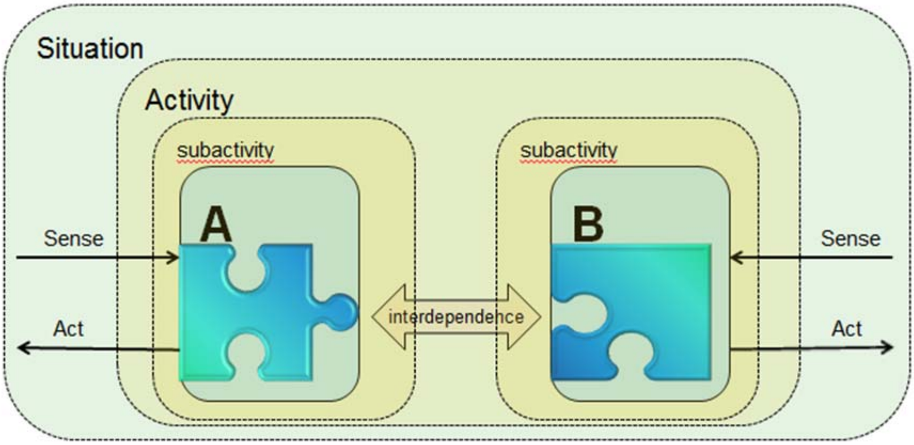



Figure 14 Context recast by interdependence

In summary, interdependence is the set of relationships used to manage dependencies. These relationships must be complementary among the parties involved. The relationships can be required or opportunistic. By engaging in such relationships, the context of the activity now encompasses all parties involved as a single joint system and these relationships then determine the available pathways to reach common ground. Our definition identifies the reasons for interdependent relationships and points to support for interdependent relationships as the remedy. However, this alone is not enough to provide the sufficiently detailed requirements necessary for implementation. Our system model will provide this missing link.

***Everything must be made as simple as possible. But not simpler.***

**– Albert Einstein**

# 7 Coactive System Model[15]

System models can be useful tools if they guide the designer to the most relevant issues to be addressed in designing a system, help to define appropriate specifications, and aid in comparing and contrasting alternatives. We will explain how our model provides simple guidelines for determining interdependence requirements. Specifically, we will describe how we propose to "manage" interdependencies and "support" interdependent relationships. In this section, we first discuss Fong's (2001) collaborative control system model, which we view as one of the best existing models of human-machine collaboration. We then propose a new coactive system model and explain how it extends Fong's model to facilitate specification of interdependence requirements.

## 7.1 Fong's Collaborative Control System Model

Fong's (2001) collaborative control model is one of the more descriptive models in the literature. In his thesis work, Fong (2001) presents a collaborative control system model, as shown in Figure 15. The role of the human in this thesis is to provide assistance to a robot that is trying to navigate (Fong, 2001). Basically, the human supplements the robot's limited perceptual and cognitive capacity. Fong's (2001) model depicts perceptual and cognitive information being provided to the human through a user interface (UI). It also depicts control input back to the perceptual and cognitive components (Fong, 2001). Fong's (2001) innovation was to suggest the human be allowed to "close-the-loop" for *both* perception and cognition. By "close-the-loop," he was referring to the making of either a perceptual or cognitive decision for a robot (Fong, 2001). An example of a perceptual decision from Fong's (2001) thesis work was answering the question, "Are these rocks?" and a cognitive decision example was answering the question, "Can I drive through?" If the questions were not answered in a timely manner, the robot would make the decision, thus this model allowed for opportunistic support, indicated by the dashed arrows for "closing-the-loop" (Fong, 2001). What enabled the distinction in Fong's system model was consideration for the internal processes of the robot, in other words, not modeling the robot as a black box. If the

[15] This chapter is adapted from (Johnson et al., 2014)

perceptual and cognitive components were not modeled, there would be no way to vary the interaction with them.

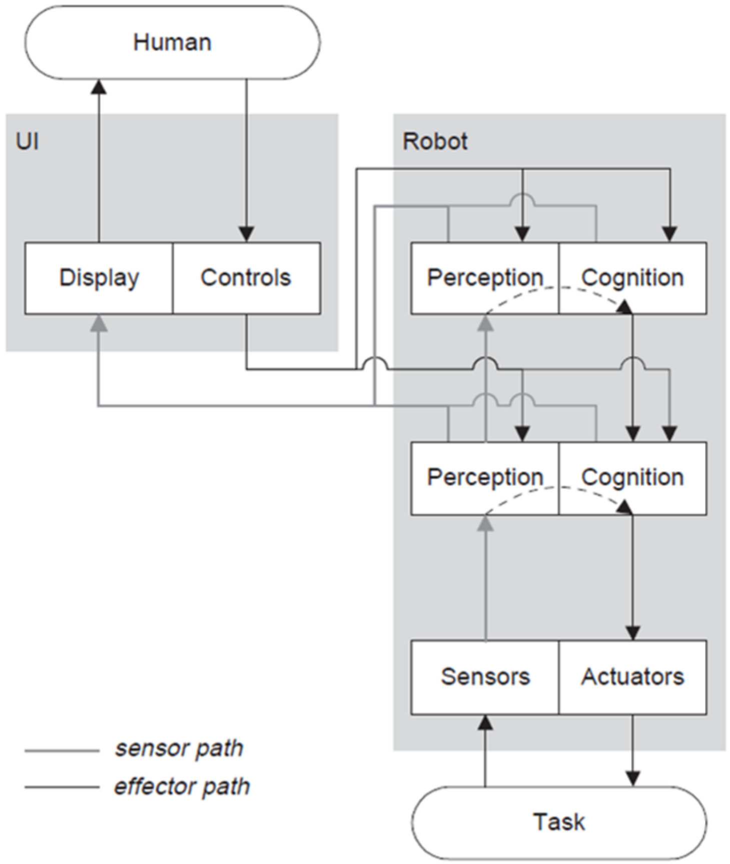



Figure 15 Fong's Collaborative Control System Model from (Fong, 2001).

## 7.2 Coactive System Model

What distinguishes joint activity from individual activity? Consider an example of playing the same sheet of music as a solo versus a duet. Clark (1996) observes that "a person's processes may be very different in individual and joint actions, even when they appear identical." The difference is that the process of a duet requires ways to support the interdependence among the players. From a designer's perspective, this means participants in a joint activity have additional requirements beyond the taskwork requirements. Where do these requirements stem from? They derive from interdependence and the need to understand and influence those engaged in the joint activity. In our framework, these requirements concern observability, predictability, and directability (OPD). The core of our system model is an abstracted interface, depicted in the middle of Figure 16. The interface captures the requirements for supporting interdependence and should shape the design of both the UI of a human operator and the robot's autonomous capabilities. We will first explain OPD and then describe the rest of the model by explaining how it extends Fong's (2001) collaborative control system model.

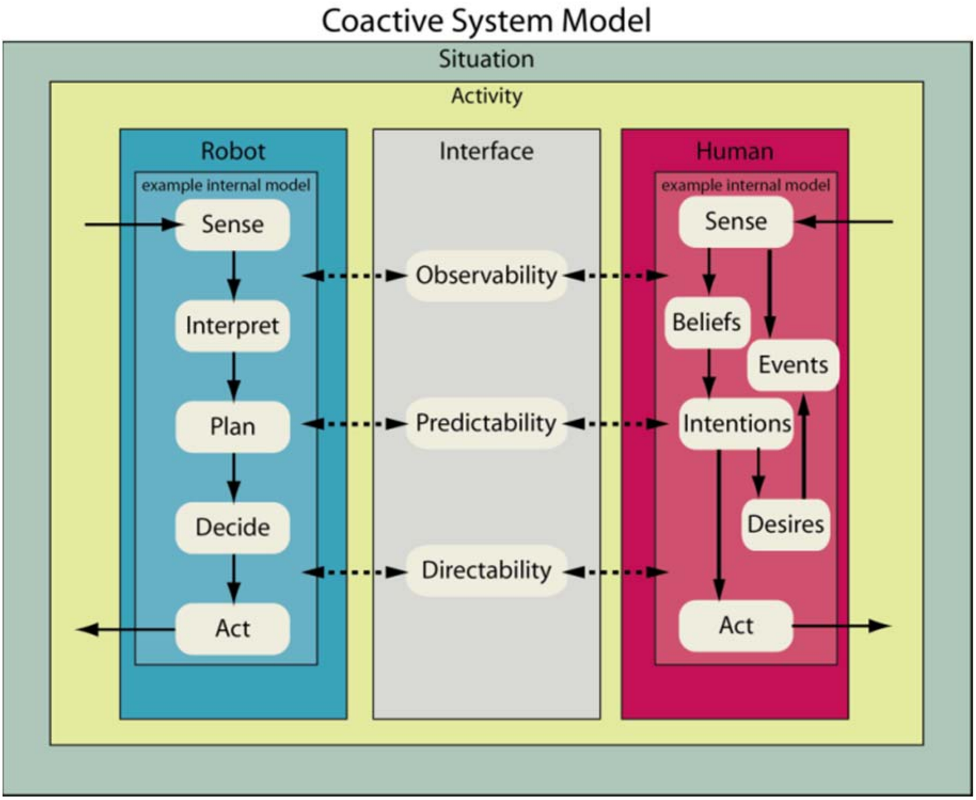



Figure 16 Coactive System Model based on observability, predictability, and directability (OPD)[16].

16 Note that we are not advocating a particular internal model for either the robot or the human. We are simply highlighting the importance of internal processes as in Clark's (1996) participatory actions and Fong's (2001) system model.

### 7.2.1 *Observability, Predictability, and Directability*

Observability means making pertinent aspects of one's status, as well as one's knowledge of the team, task, and environment observable to others. Since interdependence is about complementary relations, observability also involves the ability to observe and interpret pertinent signals. This correlates with Clark's (1996) statement that communicative acts are joint actions and his concept of joint action ladders. Though not called "observability" in the 10 challenges[17] (Klein et al., 2004), this concept aligns with Challenge 5, revealing status and intentions, and Challenge 6, interpreting signals. It also aligns with Challenge 9, attention management, which is part of Clark's joint action ladder is attention management. Observability is also consistent with work in the HRI domain (Sycara & Sukthankar, 2006), which lists team knowledge as an important facet of human-agent interaction. Observability plays a role in many teamwork patterns e.g., monitoring progress and providing backup behavior.

Predictability means one's actions should be predictable enough that others can reasonably rely on them when considering their own actions. The complementary relationship is considering others' actions when developing one's own. Mutual predictability is Challenge 3 of the 10 challenges (Klein et al., 2004) and is also listed as one of the three important facets of human-agent interaction (Sycara & Sukthankar, 2006). Dragan, Lee, and Srinivasa (2013) makes an interesting distinction between predictability and legibility, but for simplicity we will use predictability to capture both matching expectation and inference from action. Predictability may involve the use of a priori agreements, e.g., Challenge 1 of the 10 challenges (Klein et al., 2004), or it may involve the use of models, e.g., Challenge 2 of the 10 challenges (Klein et al., 2004). Challenge 2 refers to adequate models, which allows for the use of complex formal models or much simpler mechanisms, such as interface elements, which may be learned through training. Predictability is also essential to many teamwork patterns such as synchronizing actions and achieving efficiency in team performance.

Directability means one's ability to direct the behavior of others and complementarily be directed by others. Directability includes explicit commands such as task allocation and role assignment as well as subtler influences, such as providing guidance or suggestions or even providing salient information that is anticipated to alter behavior, such as a warning. Directability is Challenge 4 of the 10 challenges, although it is only described as agents being directable and does not

[17] Below we discuss all but 2 of the 10 challenges. Challenge 8, relating to a collaborative approach to teamwork and autonomy, is pervasive in coactive design and did not need special mention. Challenge 10, controlling the costs of joint activity, is not directly addressed in this paper we note that interdependence analysis helps designers to focus their attention and resources on the problems and opportunities where performance payoffs are most likely to occur.

include the complement. Challenge 7, goal negotiation, could be viewed as a type of mutual directability. Directability is also one of the important facets in human-agent interaction (Sycara & Sukthankar, 2006), although only role assignment was considered. Teamwork patterns that involve directability include such things as requesting assistance and querying for input during decision making.

Others in the HRI community have also identified OPD as critical issues. A notable example is Stubbs, Hinds, and Wettergreen's (2007) field study of HRI; they do not use the same terminology we do, but the correlation is evident. They state that "had the science team been able to observe the robot executing commands in the desert, they would have had enough contextual information to disambiguate problems" (Stubbs et al., 2007, p. 45). This is akin to observability in our model. They also state, "we noticed that issues arose around why the robot made certain decisions" (Stubbs et al., 2007, p. 47). This is an issue of predicting the robot's behavior. The system was assumed to have no directability since "only the robot could perform certain actions, and the science team couldn't exert authority in those situations" (Stubbs et al., 2007, p. 49). However, it is not hard to imagine how better support for directability would have been beneficial in the system being studied.

By using the OPD framework as a guide, a designer can identify the requirements for teamwork based on which interdependence relationships the designer chooses to support. The framework can help a designer answer questions such as "What information needs to be shared," "Who needs to share with whom," and "When is it relevant." It is important to remember that it is not just about what information you share, but also about what you do not share. Sometimes too much information can be just as big a problem. The goal of a designer is not to maximize or minimize OPD. It is to attain *sufficient* OPD to support the necessary interdependent relationships.

### *7.2.2 How the Coactive System Model Extends Collaborative Control*

Based on our definition of interdependence, we extend Fong's insightful model in several ways. The first extension is to include the human as an actor in the system model. Fong's (2001) system model is not alone in depicting the connection to a human as input and an output (e.g. control and display) and excluding the internal processes of the human (Cervin, Mills, & Wünsche, 2004; Ding, Powers, Egerstedt, Young, & Balch, 2009; Enes & Book, 2010; Fong, 2001; Michaud et al., 2010). To design for collaboration, it is essential to understand what the input and output arrows between the human and the rest of the system represent, as is the case in Figure 2. However, Fong does provide additional insight the others do not by connecting the user interface to specific internal components of the robot, specifically the perceptual and cognitive components. This implies the type of input and output that might be necessary. When we include the human in our model, it is not just as a black box or an endpoint. The human is a full actor, making coactive design considerations between partners symmetric, although the capabilities of each may not be. This means the machine could potentially "close-the-loop" for the human at any of the dimensions that compose the human's internal processes, as shown by the bi-directional arrows in Figure 16. This is more in line with the original interpretation of mixed-initiative interaction (Allen et al., 1999) than with collaborative control, which focused on a human supplementing a robot's deficiencies. This extension also means the human's potential to sense and act on the environment directly is modeled, in addition to acting through the robot, which may be appropriate for some systems. The composition of the human's internal model and that of the robot are not important to the coactive system model; composition can vary based on the designer's preference. In Figure 16, we are not advocating any particular internal models, merely providing examples to highlight the importance of internal processes as Clark (1996) points out with his duet example. Incorporating them explicitly in the model allows for inclusion in design considerations, as Fong (2001) did to enable the human to better support the robot.

The second extension to Fong's model is to include any and all relevant processes of the participant's internal model. Perception and cognition are just two of the processes that may be involved, but all processes can potentially benefit from support. It may seem odd to "close-the-loop" on sensing and acting, but people do this every day. A sensing example of support could be one person informing another about something they have noticed (e.g., "I saw the book you are looking for in my local bookstore"). An example of providing support for acting could be holding the door open for somebody, so they do not have to do it themselves. This extension also includes allowing for any permutation of "closing-the-loop." For example, sensing input is not limited to "closing-the-loop" on sensing, but may affect planning, decision making, or even the action. The plan may affect the decision or the interpretation of the new data. Fong's (2001) model could potentially mislead a designer that the cognitive processes are simple and sequential, when most activity of any complexity involves iterative framing and reframing of the problem. Our model makes no assumptions about the order of operations.

There is one more extension to the system model that fundamentally distinguishes the Coactive system model. Our model shifts the focus from individual functional components, based on supplementing capacity, to team functional components based on supporting interdependence. In essence, we decouple the individual taskwork from the teamwork. We do this by using the interface as a layer of abstraction that represents the mechanisms required to support interdependence. Here we are using interface in its general sense of a boundary between systems, as opposed to the typically graphical component for input and output commonly called the user interface. This allows different internal models of robots and humans to co-exist in the same model of the human-machine team. For example, Figure 16 shows a derivation of a standard Sense-Plan-Act model for a robot combined with a derivation of a Belief-Desire-Intention model for the human. Notice that the arrows from, for example, observability do not connect to particular parts of the robot's or the human's internal model. This is because observability may be needed to support any of the processes, such as interpretation, planning, or decision making. We show different example internal models for the robot and the human to emphasize that our model is not dependent on the underlying implementation.

In summary, this chapter describes how we "manage" interdependencies and "support" interdependent relationships. Our system model highlights three key team capabilities, over and above task capabilities, that are needed for effective human-machine collaboration: observability, predictability and directability. For team members, these three capabilities enable resilience, allowing them to "recognize and adapt to handle unanticipated perturbations" (Woods & Hollnagel, 2006). From a designer's perspective, observability, predictability, and directability are important because they provide guidance on how to identify design requirements. By determining how these capabilities must be supported in order to be capable of understanding and influencing team members, designers can create a specification. This design stance necessarily shapes not only the "user interface" for the human but also the implementation of a robot's autonomous capabilities. The shaping process is provided by the three team capabilities in our system model which capture three of the key elements required for effective teamwork.

*It is common sense to take a method and try it. If it fails, admit it frankly and try another. But above all, try something.*

**– Franklin D. Roosevelt**

# 8 Coactive Design Method[18]

Intuitively, effective teamwork implies coordination of activity, cooperation among participants and collaboration. However, all these terms are too abstract to give direct guidance to human-machine system designers and developers. The challenge is to translate high-level concepts such as teamwork and collaboration into specific requirements that can be implemented within control algorithms, interface elements, and behaviors. The result of this gap between high-level concepts and implementation is what we will call the *Gulf of Implementation*. This is similar to the issues between the user and the technology, which Norman refers to as the *Gulf of Evaluation* and the *Gulf of Execution* (Norman, 1988)[19] except it is a gap between high level concepts like teamwork and collaboration and low level implementations of such behavior.


Figure 17 The Gulf of Implementation is a gap between high level concepts like teamwork and collaboration and low level implementations of such behavior.

[18] This chapter is adapted from (Johnson et al., 2014)

[19] The gulf of execution is the degree to which the interaction possibilities of an artifact, a computer system or likewise correspond to the intentions of the person and what that person perceives is possible to do with the artifact/application/etc. In other words, the gulf of execution is the difference between the intentions of the users and what the system allows them to do or how well the system supports those actions (Norman, 1988).
The gulf of evaluation is the degree to which the system/artifact provide representations that can be directly perceived and interpreted in terms of the expectations and intentions of the user (Norman 1988). Or put differently, the gulf of evaluation is the difficulty of assessing the state of the system and how well the artifact supports the discovery and interpretation of that state (Norman, 1991). "The gulf is small when the system provides information about its state in a form that is easy to get, is easy to interpret, and matches the way the person thinks of the system" (Norman, 1988, p. 51).

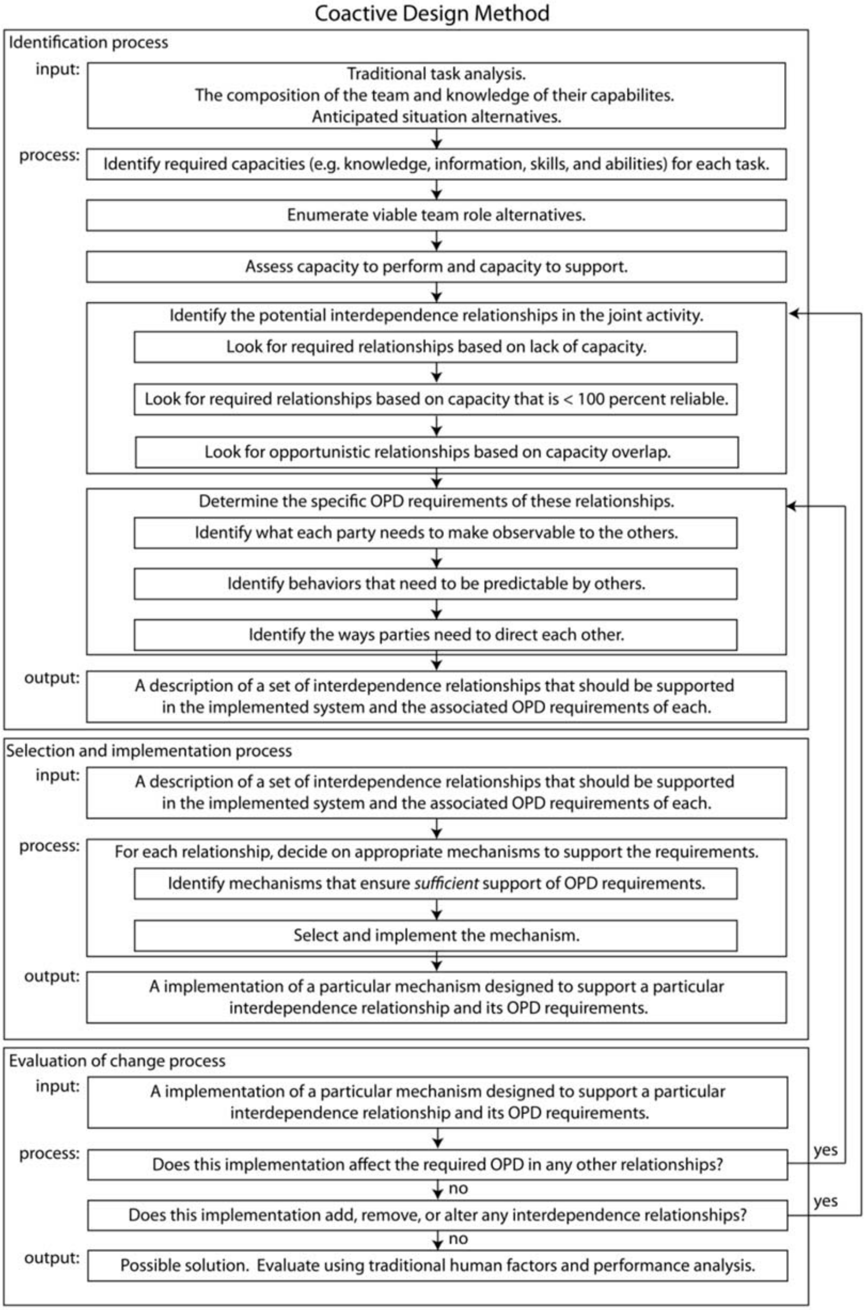



Figure 18 The Coactive Design Method

The thesis puts forth the concept that managing interdependent activity is the key design element for developing collaborative systems. It is through understanding and modeling interdependence in a human-machine system that we can provide the specification necessary to bridge the *Gulf of Implementation*. We further suggest that systems designed to support interdependence more effectively will also be better at bridging the *Gulf of Evaluation* and the *Gulf of Execution* (Norman, 1988) enabling improved performance.

With an understanding of interdependence and our system model, we can now present the general method for Coactive Design, as shown in Figure 18. There are three main processes involved in the Coactive Design method: an identification process, a selection and implementation process, and an evaluation of change process. Similar to most design processes, these will typically be iterative processes that involve feedback and refinement.

## 8.1 The Identification Process

To assist in the identification process, we propose an analysis tool that we call the Interdependence Analysis (IA) table, as shown in Figure 19. It is similar to traditional task analysis techniques (Annett, 2003; B. Crandall & Klein, 2006; M R Endsley et al., 2003; Schraagen et al., 2009), but we extend these types of analysis tools to support designing for interdependence by:

- Allowing for more types of interdependence than just task dependency
- Representing other participants in the activity by name or by role
- Allowing for assessment of capacity to perform
- Allowing for assessment of capacity to support
- Allowing for soft constraints
- Allowing for consideration of role permutations

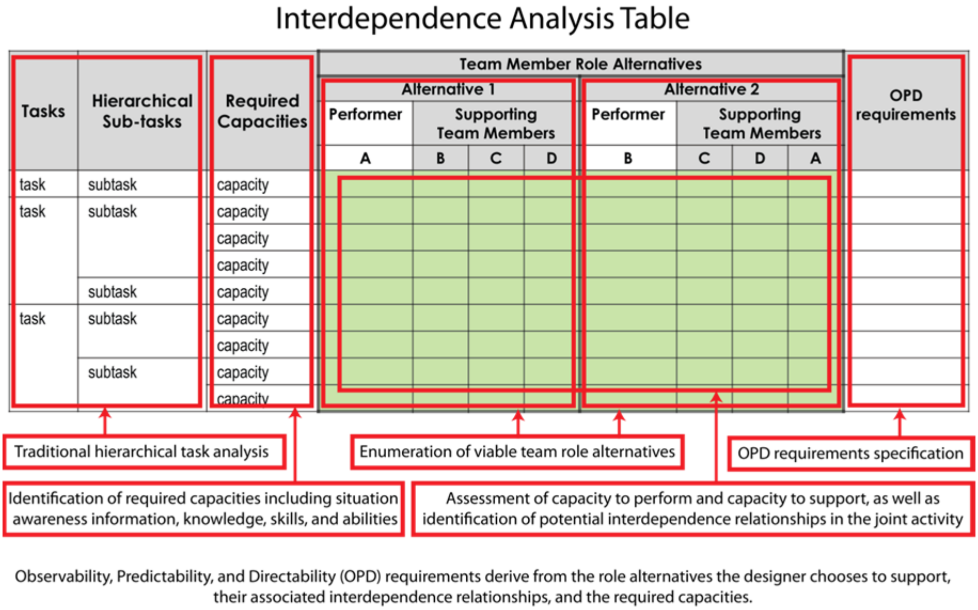



Figure 19 Explanation of the different areas of the Interdependence Analysis (IA) table

### *8.1.1 Identifying Required Capacities for Tasks*

The identification process requires a traditional task analysis as an input, as well as knowledge of the team members, their capabilities, and the anticipated situation (e.g., environment). The Left-most columns of the IA table are a traditional HTA (Annett, 2003), decomposing the task to an appropriate level of granularity. Following the HTA, we add a required capacities column to capture requirements in a manner similar to CTA (Schraagen et al., 2009) or GDTA (M R Endsley et al., 2003). However, we do not limit this to informational needs and include knowledge, skills, and abilities such as sensing needs, perception needs, decision needs, and action needs. This enables consideration for supplementing team members with any required capacity. Just as tasks may have multiple subtasks, subtasks may have multiple capacity requirements.

### *8.1.2 Enumerating Viable Team Role Alternatives*

The remaining columns are the heart of the IA. These columns enumerate the team role alternatives. They can be thought of as the adjustment options in Adjustable Autonomy or the initiative options in Mixed-Initiative. However, what they really are is an enumeration of the possible ways a team can achieve the task. A given alternative is represented by a set of columns. The first column in the set represents the primary individual performing the task. The remaining columns represent the other participants in the joint activity playing a supporting role. The columns can be specific individuals, categories, or even roles. Multiple alternatives should be analyzed by changing the performer in each alternative, as shown in Figure 19.

### *8.1.3 Assessing Capacity to Perform and Capacity to Support*

After the team alternatives are determined, the next step is the assessment. We are not looking at interdependence yet. We are just trying to understand the capacity of individual team members. To do so, we assess the individual in the column header's ability to provide the required capacity as the "performer" or to support the performer in providing the required capacity. In order to aide future analysis, the assessment process uses a color coding scheme, as shown in Figure 20. The color scheme is dependent on the type of column being assessed. The categories were chosen to help identify important system characteristics which will be discussed in the next section.

| Team Member Role Alternatives | |
|---|---|
| **Performer** | **Supporting Team Members** |
| I can do it all | My assistance could improve efficiency |
| I can do it all but my reliability is < 100% | My assistance could improve reliability |
| I can contribute but need assistance | My assistance is required |
| I cannot do it | I cannot provide assistance |

Figure 20 Interdependence Analysis Color Scheme. Note that the "Performer" column has a different meaning than the "Supporting Team Member" column.

Under the "performer" columns, the colors are used to assess the individual's capacity to do the task. The color green in the "performer" column indicates that the performer can do the task. For example, a robot may have the capacity to navigate around an office without any assistance. Yellow indicates less than perfect reliability. For example, a robot may not be able to reliably recognize a coffee mug all the time. Orange indicates some capacity, but not enough for the task. For example, a robot may have a 50 pound lifting capacity, but would need assistance lifting anything over 50 pounds. The color red indicates no capacity, for example, a robot may have no means to open a door.

Under the "supporting team member" columns, the colors are an assessment of that team member's potential to support the performer. The color red indicates no potential for interdependence, thus independent operation is the only viable option for the task. Orange indicates a hard constraint, such as providing supplemental lifting capacity when objects are too heavy. Yellow is used to represent improvements to reliability. For example, a human could provide recognition assistance to a robot and increase the reliability in identifying coffee mugs. Green is used to indicate assistance that may improve efficiency. For example, a robot may be able to determine the shortest route much faster than a human.

### *8.1.4 Identifying Potential Interdependence Relationships*

Once the assessment process is finished, the color pattern can be analyzed. The color coding scheme in Figure 20 was chosen because it provides the designer some insight in the characteristics of the system and the potential interdependence relationships. Figure 21 is not an assessment of any particular task, but a list of feasible color combinations based on our color scheme in Figure 20. Figure 21 also provides a general interpretation of the color combination. For example, if the performer is 100 percent reliable (green) and the supporting team member is not capable of providing assistance (red) then the interpretation is that the performer must meet the required capacity, whatever it may be, independently. If the supporting team member is capable of assisting (green or yellow), it might still be worth supporting, because it provides an alternative method for meeting the required capacity.

Overall, the colors in the first column provide an understanding of how the performer would fare if required to meet the capacity requirement “autonomously.” Green in the “performer: column means an autonomous approach would be fine. Colors other than green in the “performer” column indicate some limitation of the performer, such as potential brittleness due to reliability (yellow), hard interdependency due to lack of capacity (orange), or just a complete lack of capacity (red).

The “supporting team member” columns provide an understanding of what type of interdependence relationships could potentially be supported. The color red in these columns indicates that there is no chance for assistance. This makes the performer a single point of failure. If the performer is less than 100 percent reliable, you will have a brittle system. However, if you can provide support for interdependence then you can avoid the single point of failure. Colors other than red in the “supporting team member” columns indicate potential required (orange) or opportunistic (yellow and green) interdependence relationships between team members. The hard interdependencies are usually easy to identify because you cannot complete the task without it. Soft interdependencies tend to be more subtle, but provide valuable opportunities for teamwork and alternative pathways to a solution. Though it is not always possible to support every possible interdependence relationship, it is usually beneficial to support as much as time and money will allow because each provides some flexibility in the system.

These patterns suggest three guidelines for identifying interdependence relationships. The first is looking for team members who lack capacity and those that can provide it. The second is looking for team members whose capacity is not

100 percent reliable and team members that can supplement it. The third is looking for opportunistic relationships based on capacity overlap between team members.

| Team Member Role Alternatives | | *Interpretation* |
|---|---|---|
| Performer | Supporting Team Members | |
| A | B | |
| Reliable | Soft Interdependency | Independent operation by performer is a viable option, but assistance could improve efficiency. |
| | | Independent operation by performer is a viable option, but assistance could improve reliability. |
| | Must be independent | Independent operation by performer is necessary. |
| Potential Brittleness | Soft Interdependency | Performer is < 100 percent reliable, but assistance could improve efficiency. |
| | | Performer is < 100 percent reliable, but assistance could improve reliability. |
| | Must be independent | Performer is < 100 percent reliable, and no assistance is possible from this team member. |
| Missing Some Capacity | Soft Interdependency | Performer requires assistance, team member can provide it, and assistance can improve efficiency. |
| | | Performer requires assistance, team member can provide it, and assistance can improve reliability. |
| | Hard interdependency | Performer requires assistance, and team member can provide it. |
| Unachievable | | Performer requires assistance, but none is possible. |
| | | Performer cannot do task. |

Figure 21 Feasible interdependence combinations based on the IA table color scheme. The areas in the "supporting team member" columns that are not red indicate potential required (orange) or opportunistic (yellow and green) interdependence relationships between team members.

## *8.1.5 Determining OPD Requirements*

To determine the specific OPD requirements, the IA table is used to help provide a detailed specification based on who needs to observe what from whom, who needs to be able to predict what, and how members need to be able to direct each other. As an example, we have created a small IA table based on Fong's (2001) Collaborative Control work, as shown in Figure 22. In this case, the robot is capable of performing obstacle avoidance; however, it is less than 100 percent reliable in interpreting if an obstacle is passable. In Fong's (2001) example, the human was capable of providing assistance, thus increasing the reliability of the robot in this task. The requirements can be derived from analyzing the IA table in Figure 22. First we identify the alternatives we wish to support; in this case it is the human assisting with interpretation of obstacles. Next, we consider the relevant interdependence relationships. Note that task dependencies can play a role here. The human's ability to interpret depends on being able to sense the obstacle, so there is an observability requirement. Once the human has interpreted if the obstacle is passable, this information must have a way to alter the robot's behavior, so there is a directability requirement. Implied in all of this is a predictability requirement that the robot will notify the human when assistance is needed before proceeding. These particular OPD requirements are based on the desire to support a particular interdependence relationship: the human assisting in interpretation of whether an obstacle is passable. This example demonstrates how OPD requirements derive from the role alternatives the designer chooses to support, their associated interdependence relationships, and the required capacities.

| Tasks | Hierarchical Sub-tasks | Required Capacities | Team Member Role Alternatives | | OPD Requirements |
|---|---|---|---|---|---|
| | | | Alternative 1 | | |
| | | | Performer | Supporting Team Members | |
| | | | Robot | Human | |
| Navigation | Avoid obstacles | Sense obstacles | observability | | |
| | | Interpret if obstacle is passable | | | To help, the human needs to observe obstacle and be able to direct robot on interpretation. Robot must be predictable and notiify |
| | | Decide to avoid to proceed | directability | | |

Figure 22 Interdependence analysis example from Fong's (2001) Collaborative Control work, showing observability and directability requirements based on choosing to allow the human to provide interpretation assistance to the robot during navigation.

## 8.2 The Selection and Implementation Process

The selection and implementation process takes the set of relationships from the identification process and determines mechanisms that are capable of meeting the requirements. There are almost always multiple ways to address a requirement. This is a creative process that will likely remain more of an art than a science, but the OPD framework does provide evaluation criterion. The main criterion for selection is sufficiency: Does it meet the OPD requirements specified in the IA table? Other possible criteria include leveraging mechanisms across multiple relationships. For example, periodic progress updates could fill the requirement for relationships requiring knowledge of current status as well as ones requiring completion notification.

Using Fong's (2001) navigation example again, the requirements were for the robot to predictably request assistance, for the human to be able to observe the obstacle, and for the human interpretation of whether the obstacle was passable to direct the robot's behavior. His solution (Fong, 2001) was a PDA interface that would present the human with an image of the obstacle and a yes-or-no dialogue whenever there was uncertainty about an obstacle. The response in the dialogue would determine the subsequent behavior. This is clearly a sufficient solution, although one could imagine alternative solutions that still meet the requirements.

## 8.3 The Evaluation of Change Process

The evaluation of change process, in Figure 18, is critical because *the choice of mechanism can change the required OPD on other relationships as well as add, remove, or alter existing interdependence relationships, thus affecting performance*. This is a restatement of the “substitution myth” (Christoffersen & Woods, 2002), tailored to understanding the impact of design choices. The “substitution myth” concluded that reducing or expanding the role of automation in joint human-automation systems may change the nature of interdependent and mutually-adapted activities in complex ways. Our previous work demonstrated experimentally how design choices can affect performance (Johnson et al., 2012). Understanding the ways in which design choices affect the interdependent relationships is an important skill for any designer of a human-machine system engaged in joint activity. As each mechanism is implemented, it must be evaluated in the context of the entire system. This can lead to iterating through both the identification process and/or the selection and implementation process. Once an acceptable solution is reached from an interdependence standpoint, the design is ready to undergo more traditional evaluations using human factors and performance analysis.

The Coactive Design method is a starting point for designers interested in building highly interdependent systems. It was designed to be simple to follow, so it does not enumerate every caveat and nuance of the process. In future work we will provide a set of coactive design principles to aide in interpreting the method and to help avoiding pitfalls in trying to follow it.

***Simplicity is the ultimate sophistication.***

**– Leonardo Da Vinci**

# 9 Joint Activity Testbed[20]

Having presented the theory, the system model and the method, we turn to application of the ideas. The first application presented is a simulated testbed that was a catalyst for maturing both the theory and the analysis technique. Prior to the development of the Coactive Design method, the IA Table, or even the definition of interdependence, we wanted to understand the problem we were attempting to address better. We knew from the literature that there were issues with autonomy yielding its expected benefit (e.g., (Bainbridge, 1983; Norman, 1990; Woods & Sarter, 1997)). However, we wanted to uncover what relationships exist between autonomous capabilities and performance, as well as any other influencing factors. The challenge was finding an effective way to perform a controlled test in the complex domain of teamwork. We desired a type of interaction that was sophisticated enough to be interesting, yet simple enough to be clearly analyzed in great detail. So, we developed a joint activity testbed called Blocks World for Teams (BW4T). BW4T is a multiplayer game played in simulation. The game allowed for multiple human or software players in any combination. The goal of the game was for the team to find and deliver a sequence of colored boxes. This chapter presents the details of the testbed and the experimental results obtained from the testbed.

## 9.1 Introduction

It is commonly believed that increasing the autonomy of certain classes of systems will improve their performance. For example, the United States Department of Defense Unmanned Systems Roadmap (Department of Defense, 2007) states "The Department will pursue greater autonomy in order to improve the ability of unmanned systems to operate independently, either individually or collaboratively, to execute complex missions in a dynamic environment (pg. 1)." In the context of a report on a Gulf oil spill, a recent IEEE article suggested "Automation techniques will improve not only the time that it takes to do these tasks but also the quality of the results" (Bleicher, 2010).

[20] This chapter is adapted from (Johnson et al., 2012)

General conclusions of this sort can be misleading for a variety of reasons. Endsley's work (Mica R Endsley, 1999) demonstrated that failures at "higher levels of autonomy" had a worse impact and that keeping the operator involved appeared to be optimal. Enes notes that providing feedback (i.e. supporting interdependence) yielded better performance than the same control algorithm without such feedback (Enes & Book, 2010). Cuevas notes opaque indications of automation's status and behavior as a problem (Cuevas et al., 2007) and Brookshire demonstrates full autonomy as performing worse than all other conditions (Brookshire, Singh, & Simmons, 2004). In this chapter, we try to unify these issues and all issues where autonomy fails to deliver on its promise of improved performance as issues of managing interdependence. In complex joint activity involving mixed teams of humans and robots, increases in autonomy may eventually lead to degradations in performance if sufficient teamwork infrastructure to manage interdependence is not provided.

More effective management of interdependence in joint activity will become increasingly important in the coming years. The sophisticated robots envisioned for the future will be increasingly collaborative in nature, not merely doing things *for* people, but also working together *with* people and intelligent systems. Though continuing research is needed to make agents and robots more *independent* during times when unsupervised activity is desirable or necessary (i.e., *autonomy*), they must also be more capable of sophisticated *interdependent* joint activity when such is required (i.e., *coactivity*). The mention of *joint* activity highlights the need for coactive human-agent-robot systems to support not only fluid orchestration of task handoffs among different people and machines, but also combined participation on shared tasks requiring continuous and close interaction. There are examples where this type of interaction is supported and cited as being critical. Some work focused on providing observability (Drury et al., 2006; Michaud et al., 2010), other work demonstrated success by additionally including predictability (Cooper, 2007; Cuevas et al., 2007; Nielsen et al., 2007) and others included directability alternatives (Humphrey, Motter, Adams, & Gonyea, 2009; Quigley et al., 2004). Because the capabilities for coactivity interact with autonomy algorithms at a deep level, they must be embedded in system design from the beginning, not layered on with a thin veneer after the fact, as is sometimes attempted.

Based on this premise, we explore how changes in autonomy can affect various dimensions of performance when interdependence is neglected. Although our experimental results stem from a simple task domain performed in a simulation environment, both our findings in the literature on human teamwork and our experience in a variety of human-agent-robot teamwork experiments and field exercises give us reason to believe that these results eventually can be generalized.

## 9.2 Background

In Section 3.2, we described the problems of what we refer to as "autonomy-centered approaches." We conclude that:

> *Even when self-directedness and self-sufficiency are reliable, matched appropriately to each other, and sufficient for the performance of the robot's individual tasks, human-robot teams engaged in consequential joint activity frequently encounter the potentially debilitating problem of opacity, meaning the inability for team members to maintain sufficient awareness of the state and actions of others to maintain effective team performance.*

Many examples supporting this conclusion can be found in the literature. For example, Stubbs (2007) recently noted lack of transparency as a problem in human-robot interaction. More generally, this issue was identified more than two decades ago by Norman (1990) as "silent automation", and subsequently by Woods (1997) as "automation surprises." We will use the term "opacity" to highlight similar problems stemming from a lack of transparency in human-automation interaction. However, it is important to recognize that the challenges go far beyond simply not being able to see needed information. They can also involve predictability, directability or other challenges that must be addressed in order to turn autonomous systems into team players (Klein et al., 2004).

## 9.3 The Experiment

Our goal was to demonstrate that in human-robot systems engaged in joint activity, increasing autonomy without addressing interdependence may lead to suboptimal performance. All players participated by driving around a robot in a simulated world. There were two types of players. Human players manually drove (teleoperated) a robot themselves to participate as a player. We also had software agent players that could drive the robots around and provided some autonomous capabilities. The human players interacted with the artificial players through the software agent, but in general were not aware of the distinction and just thought of it as interacting with a robot. We will use the term agent player to refer to the robots controlled by software agents and which had some autonomous capability.

We attempted to rule out over-trust in automation as a failure factor by ensuring that the agent players never made mistakes and that they exhibited reasonably intelligent behavior. We also attempted to ensure that the interaction between the human and the agent could be at a relatively high level of abstraction—i.e., that the agent's capabilities for autonomy were not under-utilized. We did not want an agent capable of completing the mission autonomously managed at a low level akin to teleoperation. To this end, we provided an interface appropriate to agents' capabilities. These elements of our experimental design are illustrated in Figure 23 (A).

Figure 23 (B) illustrates the general trends we expected to find in our results. We anticipated that the management burden the agent player imposed on the human player would decrease as agent autonomy increased. Such a finding would be no surprise, since reduction in human workload is both the common expectation and the major motivation for automation. However, we also anticipated that, without support for managing interdependence issues, the opacity of the work system to task participants would grow with increasing autonomy. Due to these competing factors of burden and opacity, we expected an inflection point in team performance, where the benefits of increasing autonomy eventually would be completely offset by the negative side effects of opacity. In other words, we predicted that the highest level of autonomy would not demonstrate the highest level of team performance, consistent with the general shape of the notional bar graph shown in Figure 23 (C).

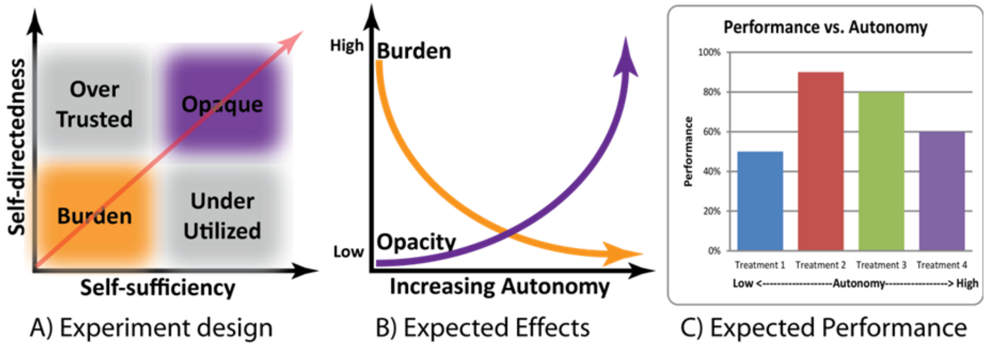



Figure 23 A) Illustration of our experimental design approach. B) Expected effects of increasing autonomy on the burden of managing the agent and the opacity of the agent to other task participants. C) Expected performance under treatment conditions of increasing autonomy, due to the competing factors of agent management burden and agent opacity.

### *9.3.1 The Experimental Domain*

Our domain for this experiment is Blocks World for Teams (BW4T) (Johnson, Jonker, Riemsdijk, Feltovich, & Bradshaw, 2009). Similar in spirit to Winograd's classic AI planning problem of Blocks World, the goal of BW4T is to "stack" colored blocks in a particular order. The task environment (Figure 24) is composed of nine rooms containing a random assortment of blocks and a drop off area for the goal. Each player controls an avatar in the game. This avatar can be moved between rooms to pick up and drop off blocks. For this experiment, teams were composed of two players—a human and a software agent. The two players work toward the shared team goal, which is to deliver the colored blocks to the drop zone in a specified order. Players are limited in their awareness of the situation: they cannot see each other and they can only see blocks that are in their current room. Human players control their own avatar in order to find and deliver blocks. They also command their agent partner through an appropriate interface. Variations on the basic game, and different experimental manipulations, can be easily programmed into the environment.

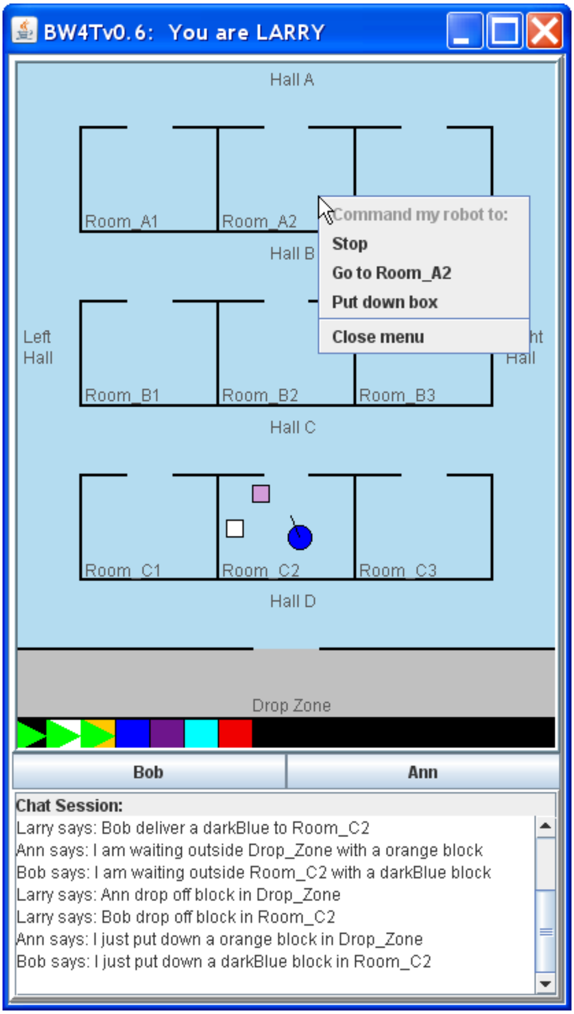



Figure 24 Example Blocks World for Teams (BW4T) interface

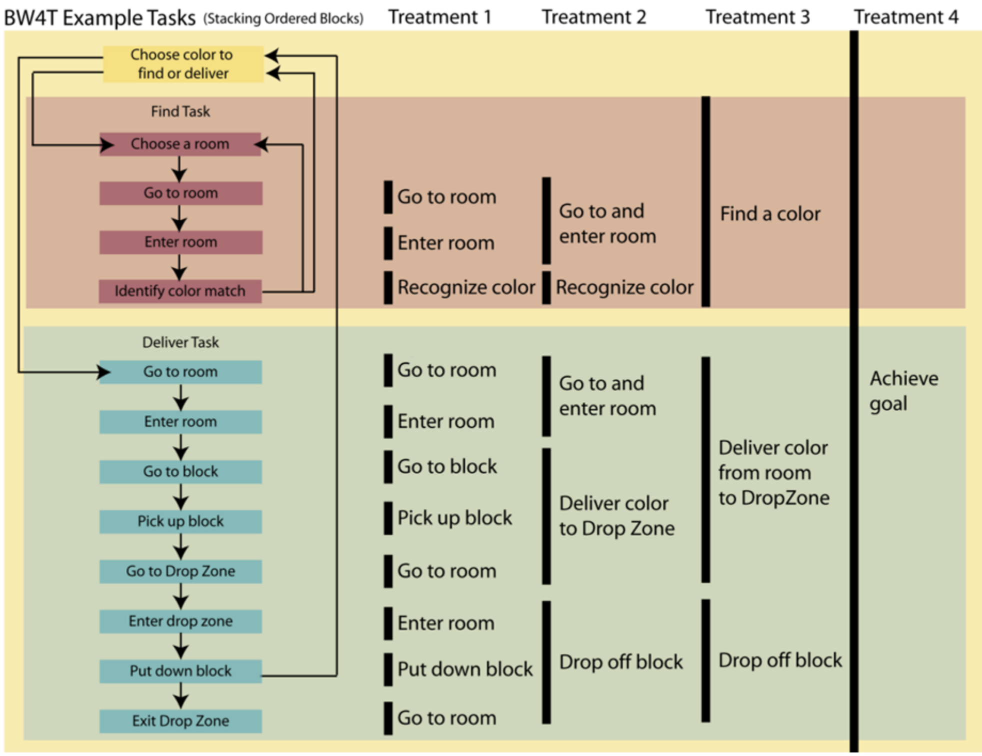



Figure 25 Defining Autonomy Treatments for BW4T

### 9.3.2 *Defining the Agent Teammate*

The algorithm chosen as the basis for the agent behavior reflects the most common approach we observed for human players of the game. This algorithm was chosen because we felt it would be easily understandable and predictable for most human players. The algorithmic solution is shown on the left side of Figure 25. The main goal (a color sequence) is composed of several subgoals (individual colors). To achieve any given subgoal, one simply finds the block of the appropriate color and delivers it. Note that these tasks need not be performed in sequence or by the same player. For example, a player could first find all the blocks and then deliver them. Alternatively, one player could find a block and another could deliver it. The overall task can be thought of as being composed of several *find* tasks and several *deliver* tasks, which are themselves composed of some decision and action primitives. The action primitives include going to a room, entering the room, going to a block, picking up a block, and putting down a block. The two main decisions are: 1) whether to look for a block or to deliver a block, and 2) which room to go to in order to look for a block. The agent player is designed to perform its task "perfectly," meaning it will perform any assigned task efficiently and will make rational decisions based on a complete and accurate recollection of where it has been and what it has seen in the past. It will also report when a task is completed. To be consistent across treatments, it *only* reports the completion status when an assigned task is completed, and does not provide any additional information.

### *9.3.3 Defining the Autonomy Treatments*

In order to compare the effects of changing autonomy, we defined different experimental conditions or "autonomy treatments." Additionally, we needed some way to rank the treatments ordinally in terms of their relative degree of autonomy. For this purpose we applied the concepts of levels of autonomy, proposed by Sheridan and Verplank (1978), and the neglect tolerance metric, proposed by Olsen and Goodrich (2003). Neglect Tolerance is a metric based on the amount of time a human can ignore a given robot performing a given task before the robot becomes unproductive.

Treatment 1 requires the human player to direct the agent player using only the action primitives. The vertical black lines or bands in Figure 25 are used to indicate the portion of the algorithm that is performed autonomously by the agent player. During the time it spends in the black band, the agent can be considered as functioning at Sheridan's highest level of autonomy, since the agent will perform on its own everything necessary to complete the task specified by the band. Outside the band, the agent is at the lowest level of autonomy and is completely reliant on the human for all decisions and actions. The behavior associated with each band is always initiated by the human teammate. The neglect tolerance correlates to the length of the band, though the band covers a portion of the algorithm and does not directly correspond to length of time, since some tasks take longer than others. However, longer bands cover more sections of the algorithm; thus, in general they entail more autonomy. The bands in treatment 1 are the shortest, requiring more direction from their human teammate and therefore have the lowest neglect tolerance.

In Treatment 2, we combine several action primitives into a single action. For example, with a single command the agent can now be ordered to go to and enter a room. To inhibit under-utilization, the command set available to the human player was restricted to the new "higher-level" commands listed under treatment 2 in Figure 25. We are only combining action primitives, so Sheridan's scale does not provide much guidance, but it is clear that agent neglect tolerance increases and, thus, this treatment has more autonomy for the agent than the first.

Treatment 3 extends Treatment 2, further combining activities and provides the ability to command the agent to find a color. This new command delegates the decision on where to search to the agent, who is now required to provide its own search algorithm and only reports when a color is found. This was implemented as a nearest-unsearched room algorithm. Prior to this experiment we ran some human-human teams to better understand the requirements for designing our software agent players. The nearest-unsearched room algorithm was chosen because it was the most common approach for human players. Again, the human player was restricted to the commands that are listed under treatment 3 in Figure 25. Consistent with Sheridan's specification for levels of autonomy, this is a higher level of autonomy than the previous treatment, since the agent can now make its own decision on how to achieve the find task. The level of neglect tolerance is also higher.

Treatment 4 extends Treatment 3, combining all activity allowing the agent to choose whether to look for a block or deliver a block. This enabled the agent player to be able to complete the entire task without any assistance from the human player. This competence level equates to Sheridan's highest level of autonomy and an infinite tolerance for neglect. The only required command by the human player is to tell the agent to achieve the goal. As in Sheridan and Verplank's level ten, the agent "decides everything, acts autonomously, ignoring the human" (Parasuraman et al., 2000).

We have intentionally left out any support for managing interdependence, except for communicating task completion status. There is neither communication about world state nor coordination of task activity. While this may seem extreme in this simple domain with obvious coordination needs, we believe it is not unrealistic given the prevalence of similarly opaque systems (Norman, 1990; Stubbs et al., 2007; Woods & Sarter, 1997). By this means, we hoped to explore the relationship between autonomy and interdependence.

### *9.3.4 Experimental Design*

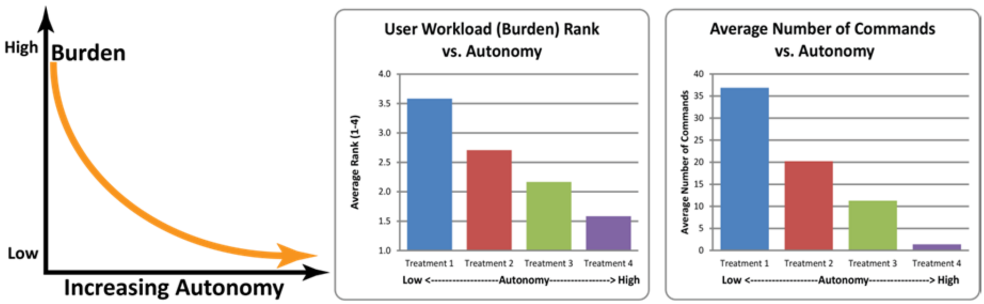
24 participants (17 male and 7 female) were selected from a student population at TU Delft, with an age range of 19-39. We employed a complete randomized block design based on the autonomy treatment, with each participant performing each treatment once. The data are cross-classified by $k = 4$ autonomy treatments and $b = 24$ blocks, consisting of the individual participants. All participants received a demographic survey. They were trained on the game until they demonstrated proficiency by completing a simplified version of the task. Next they performed a series of trials, one for each treatment. The participant filled out a brief survey at the end of the experiment, evaluating team burden, opacity, performance, and preference in each treatment.

## 9.4 Results

Our results include quantitative numeric data as well as subjective ranking data. For the former, we use standard approaches for normal data. For the ranked data, we used the nonparametric Friedman test. Based on our design, and using the $\alpha$ = 0.05 level of significance, the critical value is $\chi^2_{.95,3} = 7.815$.

### 9.4.1 *Assessing Burden*

Our hypothesis predicted a decrease in agent management burden as autonomy increased from treatments 1 to 4. This is depicted in Figure 26(A). We asked the participants to rank how demanding it was to work with the agent in each condition, on a scale of 1 (least demanding) to 4 (most demanding). The results, shown in Figure 26 (B), indicate a very clear ($\chi^2_{.95,3}$ = 34.225) decrease in burden as autonomy increased. As a second, independent measure of burden, we also counted the number of commands the human player had to give to the agent teammate in each condition. Figure 26 (C) shows the results, which correlate with the subjective assessment.




A) Expected change in burden B) Qualitative burden measure C) Quantitative burden measure

Figure 26 (A) Expected change in burden as autonomy increases (B) Subject ranking of agent management workload (burden) as autonomy increases across experimental treatments. (C) Average number of commands (Burden) as autonomy increases.

### 9.4.2 *Assessing Opacity*

Our hypothesis predicted an increased subject perception of opacity with increasing autonomy across the experimental conditions. This is depicted in Figure 27 (A). We expected this to be reflected in reports of subjects having more difficulty in understanding what was happening and in anticipating the agent's behavior as autonomy increased. An exit survey was used where subject were asked to rank their ongoing sense of awareness of current and future agent actions in the different conditions on a scale of 1 (most aware) to 4 (least aware). The results in Figure 27 (B) show opacity increasing with increasing autonomy as predicted ( $\chi^2_{.95,3}$ = 49.700). This confirms our prediction about opacity in this experimental setting, and validates the general expectation.

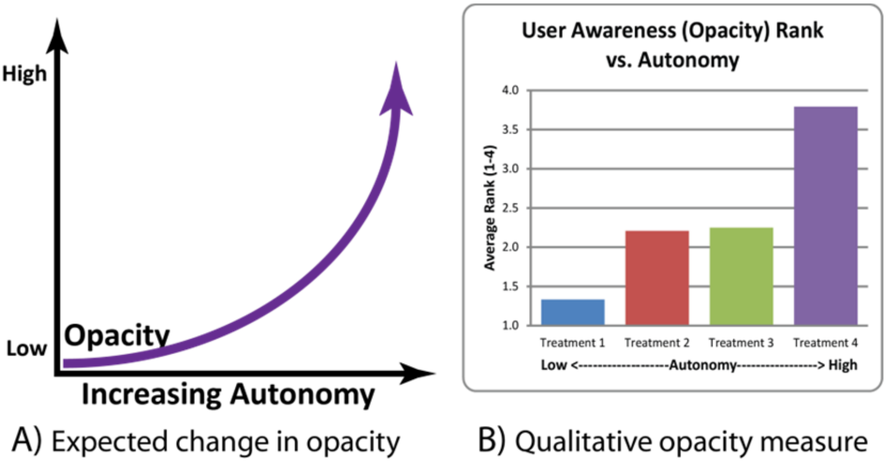



Figure 27 (A) Expected change in opacity as autonomy increases (B) Average subjective ranking of awareness (opacity) as autonomy increases across experimental treatments.

### *9.4.3 Quantitative Performance Assessment*

We performed three different quantitative performance assessments: time to complete task, idle time, and error rate.

#### ***9.4.3.1 Time to complete task***

The simplest performance metric is time to completion—i.e., delivering all the required blocks in the requested order. Figure 28 shows the results. At first glance, the results appear promising. We can clearly see the inflection point where performance begins to degrade rather than improve under conditions of increasing autonomy, consistent with the prediction of Figure 23 (C). The differences, however, were not statistically significant (p = 0.20). We believe that this is best explained by the fact that the task itself has a large amount of variance from run to run, and the penalty incurred by errors is less than the variance between runs. We note, however, that in 83% of the participants, the highest-autonomy condition (Treatment 4) was not the highest-performing condition by the time-to-completion criterion.

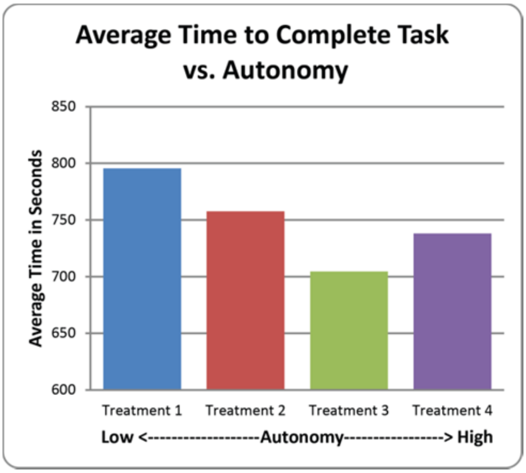



Figure 28 Time-to-completion as autonomy increases across treatments.

#### *9.4.3.2 Idle Time*

Another important performance measure is idle time (or wait time (J. W. Crandall & Cummings, 2007)). In the BW4T task, the agent player will be in near constant motion once a task has been assigned to it by its human teammate. Any idle time is indicative of inefficient use of the agent player (e.g., while it awaits the next command). Figure 29 (A) shows the results of average idle time for the agent player. There is a clear and significant decrease in idle time from treatment 1 to 4. On the surface, this could be taken as indicating more effective use of the agent player by the human, and thus suggesting improved performance. However, this is not borne out by the time-to-completion results (Figure 28). Additionally, we note that the amount of work done is fairly consistent across treatments. For example, the number of rooms entered and the number of boxes delivered does not change much across treatments. This also makes sense when one looks at the human player's idle time, shown in Figure 29 (B). There is a slight decrease in idle time as the burden is reduced, but not much, and certainly not on the order of the change seen in the agent player. This indicates that the interaction efficiency (J. W. Crandall & Cummings, 2007) is not that significant. This could be due to an effective interface, but it also can be due to the ability to multi-task and complete interactions concurrent with motion. The interesting takeaway lesson from this result is that "keeping your agent busy" does not equate to improved performance.

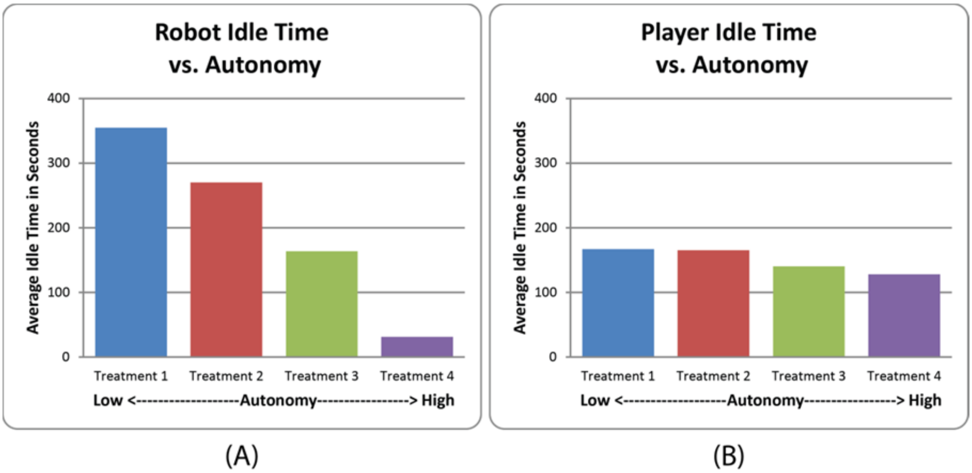



Figure 29 (A) Average agent player idle time across treatment conditions. (B) Average human player idle time.

#### *9.4.3.3 Error Rate*

For some kinds of tasks, error rate can be a good way to compare performance. We measured this in three ways. Our first was the amount of time that both players spent holding the same color block (Figure 30 (A)). Since, for this experiment, the goals were composed of unique colors (no repeats), this represented a measure of some fraction of overall redundant activity or inefficiency in task performance. This type of error, for the most part, only occurred in treatment 4 and is a side effect of the high opacity of the highest-autonomy condition. These results are no surprise, since this is the only treatment in which the agent player can make its own decision about which block to pick up. However, this does emphasize that functional differences matter when automating tasks (Johnson, Bradshaw, Feltovich, Hoffman, et al., 2011).

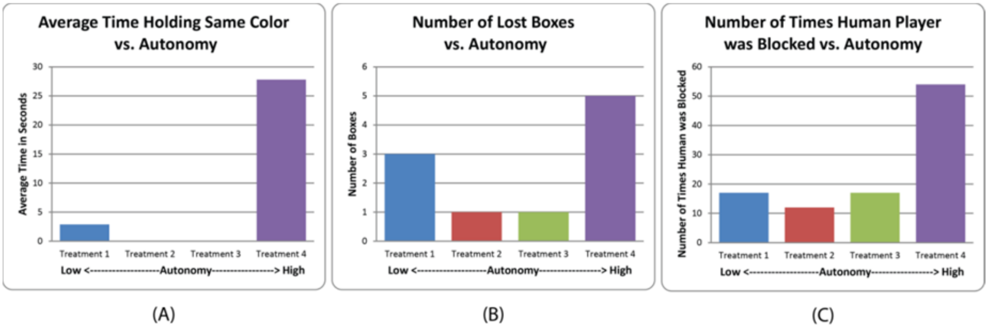



Figure 30 (A) Average time holding the same color (inefficiency) (B) Number of lost boxes (C) Number of times a human player was blocked by their agent partner while trying to enter a room.

A second measure of error is the number boxes lost—i.e., dropped in the hallway or placed in the drop zone erroneously. Since BW4T is very simple, there were not many mistakes made by the human players, but of the ten lost boxes, 50% of them occurred in treatment 4 and 30% occurred in treatment 1, as shown in Figure 30 (B). The boxes lost in treatment 1 were most likely due to the high workload imposed by the minimal amount of autonomy. However, treatment 4 does not have the obvious workload challenges of treatment 1. In fact, it was clearly ranked as the least burdensome, so why would it have the highest occurrences of errors? We believe the high error rate is a side effect of the high opacity of the highest-autonomy condition.

Our third measure of error was the number of times a player was blocked while entering a room. This measure is indirect because it is possible that the most efficient act would be to wait outside a blocked door, but in general it indicates poor coordination. As shown in Figure 30 (C), the human player was blocked in treatment 4 much more often, indicating significantly more coordination breakdowns than any other treatment.

### *9.4.4 Subjective Performance Assessment*

#### ***9.4.4.1 User Performance Assessment***

We asked the subjects to identify which team they felt performed best. Treatment 3 was the clear winner, with 63% of the participants selecting it as the best performing treatment (Figure 31(A)). Only 17% of the subjects choose treatment 4 as the best performing.

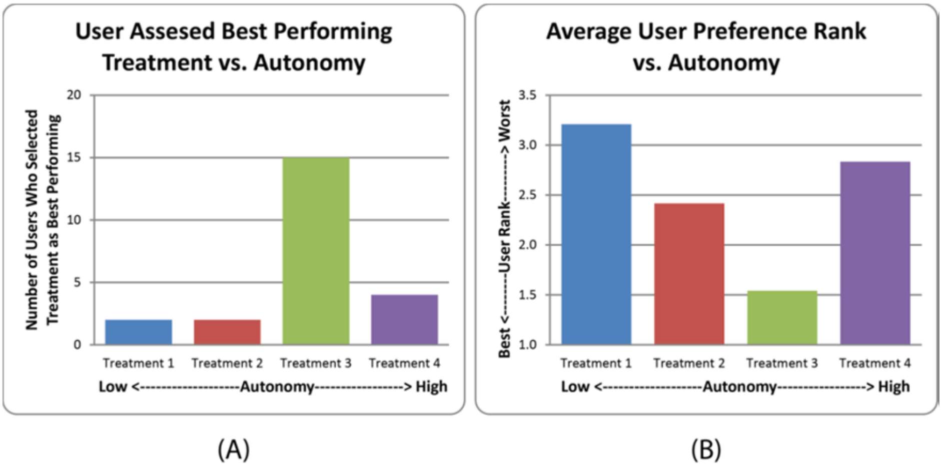



Figure 31 (A) User Assessment of Performance vs. Autonomy (B) User Preference vs. Autonomy.

#### *9.4.4.2 User Treatment Preference*

Human acceptance is an important component of overall system performance in tasks like ours. We asked the participants to rank the agents in each experimental condition with respect to their preference as to which one they would like to play with again, on a scale of 1 (most like to play with again) to 4 (least like to play with again).

Figure 31 (B) shows the results. Treatment 3 was preferred with statistical significance ($\chi^2_{.95,3} = 22.150$). This result also demonstrates the inflection point anticipated by the increasing opacity in the system from Figure 23 (C). We suspect this is because in treatment 3 the human holds the overall plan, most of the context, and exercises the greatest degree of creativity. In this context, transparency and control (directability) may be more important than autonomy (independent operation), especially in light of the particulars of the autonomous task.

We asked participants about the reasons for their rankings, and the responses were enlightening. Reasons for preferring Treatment 3 included:

- Shared information
- Able to anticipate
- Predictable
- Low burden
- Cleverest

- Automatic, but still have control

The first three reasons correlate with our predictions about opacity. The comment about low burden is interesting, because treatment 4 was objectively less burdensome. This comment suggests that there may be other types of burden besides the manual workload of tasking the agent. The comment about treatment 3 being cleverest is also interesting, because treatment 4 is objectively the most capable (clever) based on what the agent can do on its own (Figure 25). Perhaps this suggests that sometimes being more independent may not necessarily lead to being viewed as more clever. The final reason is also important because it relates to the broader issue. We focused on opacity in order to keep the experiment simple, but predictability, directability and other challenges in making automation a team player (Klein et al., 2004) are no doubt also affected by increased autonomy.

## 9.5 Conclusions

The results of our initial limited evaluation support our claim that increasing autonomy does not always improve performance of the human-machine system. In our example, increasing autonomy improved performance up to a point, but then there was an inflection point where performance decreased, depicted in Figure 32. We saw performance inflections in time, in error rates, and in user rankings. We propose that systems that fail to address interdependence adequate with have similar inflection points in performance. In the BW4T domain, this was principally due to opacity in the system, derived from increasing autonomy without accounting for the interdependence of the actions and decisions of the players and the coordination challenges this creates. Additionally, we showed how keeping an agent busy does not equate to improved performance, how human error rates are not only due to workload but can also be affected by opacity, and how user preference is not necessarily driven by reduced burden when other factors such as transparency, predictability and directability are relevant to the task. A key point to take away is that the ability to work *with* others becomes increasingly important as interdependence in the joint activity grows. It is possible that in complex and uncertain domains, this may be more valuable than the ability to work independently.

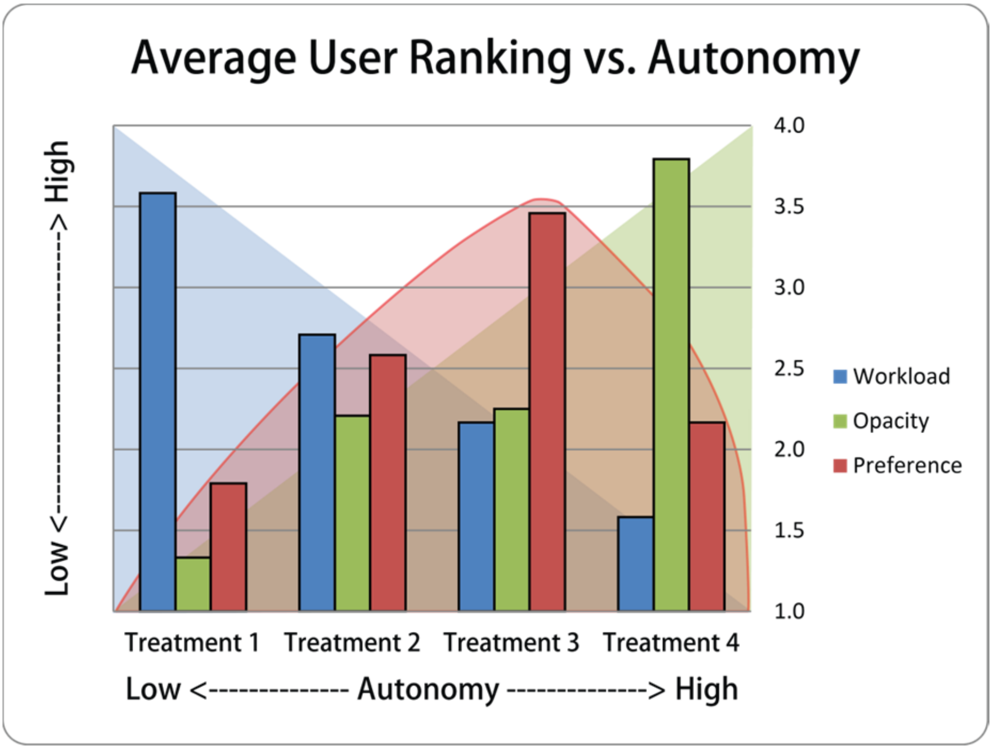



Figure 32 Performance inflection point demonstrated by results

It is obvious why opacity has such an effect on the system in the BW4T domain. The greater the autonomy of players, the greater the opacity, and hence the more room for coordination breakdowns. The independent activity in treatment 4 inhibited the team's ability to engage in what most people would consider "natural" coordination, resulting in a breakdown of common ground (Klein et al., 2005) and reduction in each player's individual situation awareness. This then caused suboptimal decisions and errors. While obvious in this simple, abstract domain, the problem remains prevalent in many systems today, as noted by several researchers (Norman, 1990; Stubbs et al., 2007; Woods & Sarter, 1997). Understanding the relationship of autonomy to interdependence is one step toward addressing the challenges facing future systems. We believe that consideration for interdependence while designing the autonomous capabilities of an agent can mitigate the effects demonstrated and will enable future systems to achieve greater potential.

***Design is the application of intent –***
***the opposite of happenstance, and an antidote to accident***

**– Robert L. Peters**

# 10 Applying Coactive Design to UAV Navigation[21]

Early in the genesis of Coactive Design we began an unmanned aerial vehicle (UAV) project. Its purpose was to demonstrate how a system designed from a coactive perspective could be more successful than a traditional one. The overall goal was to demonstrate effective navigation through obstacles, which remains a challenging endeavor for current systems. This is a task that is difficult for either humans or unmanned vehicles to currently complete successfully on their own in situations of any significant complexity — harnessing the capabilities of each in effective teamwork is required. This was accomplished not by guessing at what widget or feature might be useful, but by an intentional approach to support interdependence among the human operator and the UAV through mechanisms that allowed the operator to coactively participate in navigation.

## 10.1 Introduction

The Unmanned Systems Roadmap (Department of Defense, 2007) stated that "The single most important near-term technical challenge facing unmanned systems is to develop an autonomous capability to assess and respond appropriately to near-field objects in their path of travel." In other words, obstacle avoidance is a critical problem for unmanned systems. Micro Aerial Vehicles, or MAVs, exacerbate this challenge because they are likely to be deployed in environments where obstacle-free flight paths can no longer be assumed. This poses a tremendous navigation challenge to such small platforms that have limited payload and sensing capability.

Teleoperation is a common mode of operation for unmanned systems, but is challenging for a variety of reasons including the limited field of view, poor situation awareness and the high operator workload. Autonomy has its own challenges in developing robust sensing, perception and decision making algorithms. Higher levels of autonomy are being vigorously pursued, but paradoxically, it is also suggested that these systems be increasingly "collaborative" or "cooperative" (Department of Defense, 2007). These terms are difficult to define—and even more challenging to map to engineering guidelines.

---

[21] This chapter is adapted from (Johnson, Carff, et al., 2012)

So, we come to the question: exactly what makes a collaborative or cooperative system? We suggest that support for interdependence is the distinguishing feature of collaborative systems and that effectively managing interdependence between team members is how teams gain the most benefit from teamwork.

The basic premise of our approach is that the underlying interdependence of the joint activity is the critical design feature, and is used to guide the design of the autonomy and the interface. To demonstrate Coactive Design for human-MAV team navigation we used the ArDrone, shown in Figure 33, as our example MAV. The ArDrone is an inexpensive commercial vehicle. It has a low resolution (640x480) forward facing camera with a 93 degree field of view, an onboard inertia measurement unit and a sonar altimeter. It also has downward facing camera that it uses for optical flow to determine velocity and localize itself. While there are more capable platforms available, we chose this one to highlight the effectiveness of our approach even when using a platform with limited sensing and autonomous capabilities and we feel it is representative of the type of systems in use today.


Figure 33 ArDrone

The environment was designed to mimic challenges expected in urban environments and included features similar to windows and doors, as well as obstacles such as walls, boxes, power lines, overhangs, etc., as would be found in typical urban areas. Figure 34 is an example of several obstructions that must be navigated and a window that must be entered.

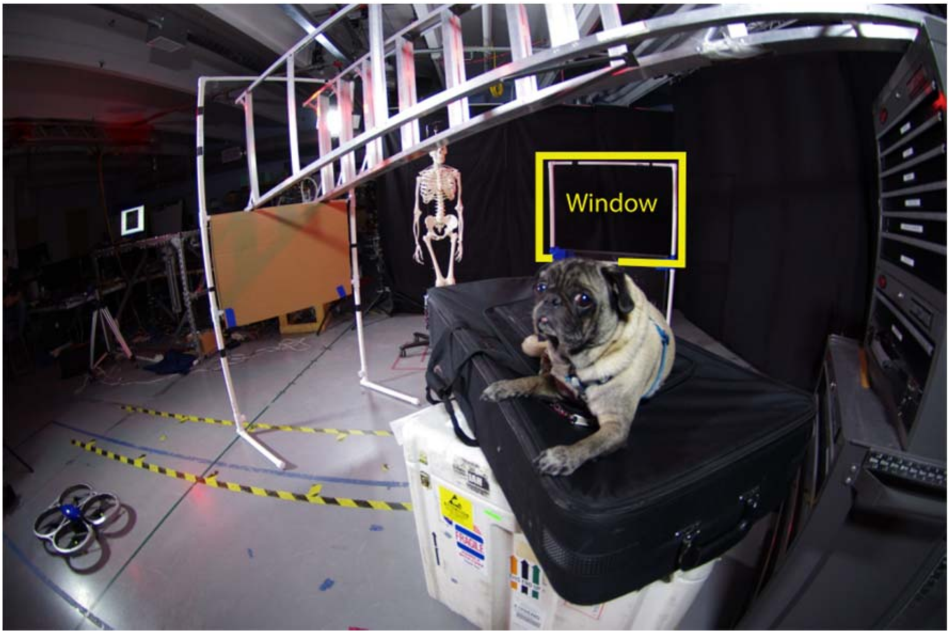



Figure 34 Example of obstacles used to evaluate the system. The obstacles would be arranged to create different challenges for the operator. Passing safely through a particular window was a typical navigation goal.

We employed our Coactive Design approach to develop a human-MAV team system capable of navigation and obstacle avoidance in complex environments. We present this system and demonstrate its unique capabilities.

## 10.2 State of the Art

Today's deployed UAVs do not have obstacle avoidance capability (Department of Defense, 2007) and this prevents their use in many important areas. The standard control station for small UAVs is composed of a video display and some joysticks for teleoperation, similar to the one shown in Figure 35. These interfaces place a high burden on the operator.

Figure 35 Teleoperation interface from IMAV 2011 competition

Systems that rely on autonomy typically only provide an overhead map view. The ground control interface provided by Paparazzi (Brisset & Hattenberger, 2008), shown in Figure 36, is a popular example and was used in the International Micro-Aerial Vehicle (IMAV) competition in 2011.

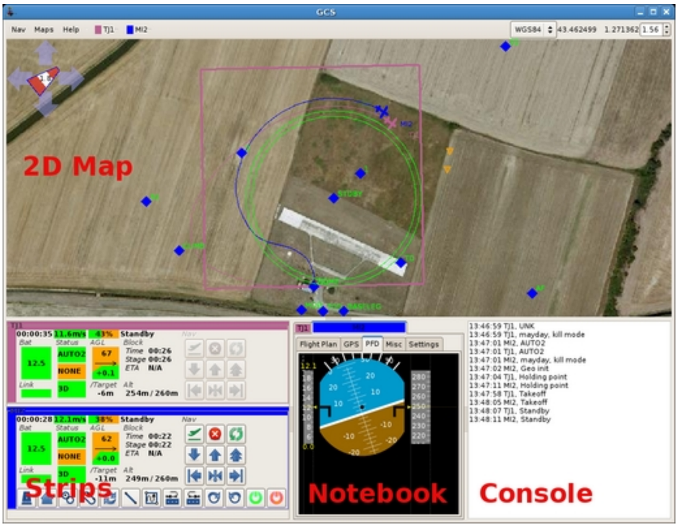



Figure 36 Paparazzi Ground Control Interface (Brisset & Hattenberger, 2008)

In some systems the two approaches are combined in a display that presents a 2D overhead map and a live video feed. However, there is no connection between the video and the map and the operator is required to perform the cognitive association between the two displays, which makes context switching difficult and error prone.

Even more important, the operation of the vehicle is viewed as a binary decision: either the vehicle is autonomous or the operator is flying. This is commonly accomplished by literally flipping a switch on a controller similar to the one in Figure 35. The transition between the two modes is often chaotic and a high risk activity. There is no collaboration. Neither the human or machine can assist the other in any way.

Each of the approaches described above has its own challenges and limitations. Many researchers have been investigating interesting ways to overcome these limitations. Some have investigated better ways to present video and map information (Cervin et al., 2004; Cooper, 2007; Drury et al., 2006; Lorite et al., 2013) others have investigated better control methodologies (Quigley et al., 2004) and others have proposed design approaches aimed at targeting these types of issues (Adams et al., 2009; Cooper, 2007; Cummings et al., 2007; Goodrich, 2004). We leverage this work and extend it to include the additional challenge of avoiding obstacles.

## 10.3 Our Approach

Our approach is about designing a human-machine system that allows the two to perform as a team, collaboratively assisting one another. We do not try to simply allocate the task of navigating to the human or the machine, but involve both in the entire process. As such, there are no modes and therefore there is no transition or handoff between the human and machine.

The basic premise of our approach is that the underlying interdependence of the joint activity is the critical design feature, and is used to guide the design of the autonomy and the interface. Anybody who has developed or worked with a robotic system has at one time or another asked questions like “What is the robot doing?”, “What is it going to do next?”, or “How can I get it to do what I need?” These questions highlight underlying issues of observability, predictability and directability which are consistent with the ten challenges of making automation a team player (Klein et al., 2004). Interestingly, addressing these issues is much more about addressing interdependence then it is about advancing autonomy. From this perspective, the design of the autonomous capabilities and the design of the interface should be guided by an understanding of the interdependence in the domain of operation. This understanding is then used to shape implementation of the system, thus enabling appropriate coordination with the operator. We no longer look at the problem as simply trying to make MAVs more autonomous, but, in addition, we strive to make them more capable of being interdependent. So how does this apply to MAV operations in complex environments?

Instead of taking an autonomy-centered approach and asking how to make a MAV that can meet this challenge autonomously, we consider the human-machine team and ask how the system as a whole can meet this challenge. More specifically, how we can meet the challenge while minimizing the burden on the human. When thought of as a joint task, we have a lot more options. We still have the options of "full" autonomy and complete teleoperation, but these are not as attractive as the middle ground. This is evidenced by the large body of work on various forms of adjustable autonomy and mixed initiative interaction (Allen et al., 1999; Bradshaw, Feltovich, et al., 2004; J. W. Crandall & Goodrich, 2002; Dias et al., 2008; Kortenkamp, 2001; R. Murphy et al., 2000) including the Technology Horizon's report (United States Air Force, 2010) which calls for "flexible" autonomy. While it is important for the autonomy to be flexible, we feel it is even more important to take a teamwork-centered (Bradshaw et al., 2004) approach. Coactive Design is such an approach.

## 10.4 Interdependence in the Navigation Domain

Interdependence in the navigation task can be understood in the context of the abilities required to successfully navigate. Figure 37 is an Interdependence Analysis (IA) table for the activity.

| Tasks | Required Capacities | Team Member Role Alternatives: Alternative 1: Perform | Alternative 1: Support | Alternative 2: Perform | Alternative 2: Support | OPD Requirements |
|---|---|---|---|---|---|---|
| | | Robot | Human | Human | Robot | |
| Sensing | Sense UAV position | | | | | Robot position estimate error varies and grows over time. Human's situation awareness is hampered by the limited field of view. |
| | Sense obstacles in the environment | | | | | Robot cannot recognize obstacles. Human's situation awareness is hampered by the limited field of view |
| Interpreting | Interpret obstacles and openings as passable or not | | | | | Robot has no ability to interpret obstacles. Human can easily interpret obstacles |
| Planning | Understand vehicle width in relation to obstacles | | | | | Robot can requires knowledge of obstacles. Human's accounting for width is not reliable. |
| | Understand appropriate standoff ranges from obstacles | | | | | Robot can not interpret standoff range requirements. Human can but could benefit from graphical aides. |
| | Plan path that does not collide with obstacles and has appropriate standoff ranges | | | | | Robot can accurately plan if provided obstacles and stand off ranges. Human can do this, but is not reliable nor accurate. |
| Execution | Track position and heading of planned trajectory | | | | | Robot can do execute well. Human is not reliable nor accurate, particularly from first person view |

Figure 37. UAV navigation Interdependence Analysis table.

Using the IA table it is straight forward to identify challenges from both the human and machine perspective. These challenges are listed in the second column of Table 4.

Table 5 Some of the remote navigation challenges for both teleoperation and full autonomy and the opportunities that are possible by taking a Coactive Design perspective.

| Required Abilities | Challenges | Opportunities |
|---|---|---|
| Sensing | Robot's onboard sensing errors | Enable human correction of deviations |
| | Human's situation awareness is hampered by the limited field of view | Enhance the human's field of view through advanced interface design |
| Interpreting | Robot's poor perceptual ability | Human's excellent perceptual ability |
| | Human's assessment of robot's abilities may be inaccurate | Provide insight into robot's abilities |
| Planning | Robot's planning is only as good as the known context | Enable human to assist with context and judgment |
| | Human's precision may be inadequate | Provide visual feedback to the human |
| Execution | Robot's navigational errors | Provide insight into how the robot is performing |
| | Human's precision may be inadequate and is limited to a first person perspective | Provide multiple perspectives to improve human performance |

Sensing involves the acquisition of data about the environment. For remote operation, the human is limited by the available sensors presented in the interface. Typically this is a video, with a limited field of view. Often operators refer to remote operation as "looking through a soda straw." In a standard interface the human operator is restricted to this single point of view and must maintain a cognitive model of the environment in order to reason about things outside of this limited field of view.  The MAV is also limited by the accuracy of its knowledge. All vehicles have onboard sensing error, so the data it senses will be subject to this error.

Interpretation of video scenes remains an open challenge for autonomous vehicles. While some successes have been made, these systems remain very fragile and highly domain dependent. The human ability to interpret video is quite amazing, but the operator must cognitively interpret vehicle size and handling quality as well as other important things such as proximity to obstacles.

Planning is something machines do well, but the plans are only as good as the context in which they are made. Great planning ability is useless without accurate and complete sensing and interpretation. Machines also lack the judgment faculties of a human. While humans can also plan well, the plans tend to be imprecise.

Machine execution is generally better than human execution for well-defined static environments. Machines are more precise and their performance is highly repeatable. However, they are limited by all the preceding abilities, such as onboard sensing error and poor perceptual abilities. Human operators are limited by their skill level and the interface provided.

While each of the challenges listed in the second column suggest difficulty for either a teleoperated solution or an autonomous solution, they also suggest opportunities, listed in column three of Using the IA table it is straight forward to identify challenges from both the human and machine perspective. These challenges are listed in the second column of Table 4.

Table 5. The Coactive Design approach takes advantage of the opportunities by viewing the navigation task as a participatory (Clark, 1996) one for both the human and machine. Individual strengths are not an indication of who to allocate the task to, but an opportunity to assist the team by providing an appropriate teamwork infrastructure. Weaknesses no longer rule out participation, but suggest an interface that supports assistance to enable all parties to contribute.

## 10.5 Our Interface

To address the observability requirements identified in Figure 37 we developed an interface to assist the human's situation awareness. Our interface, shown in Figure 38, is composed of a 3D world and two views into that world. The left view is the view into that world from the perspective of the MAVs camera. The right view is an adjustable perspective with viewpoint navigational controls similar to Google Earth. Similar use of 3D style interfaces has proven to be effective, particularly in the navigation domain (Michaud et al., 2010; Nielsen et al., 2007). We provide a few control buttons and a battery level, but in general our interface is devoid of gauges and dials that typically clutter unmanned system interfaces.

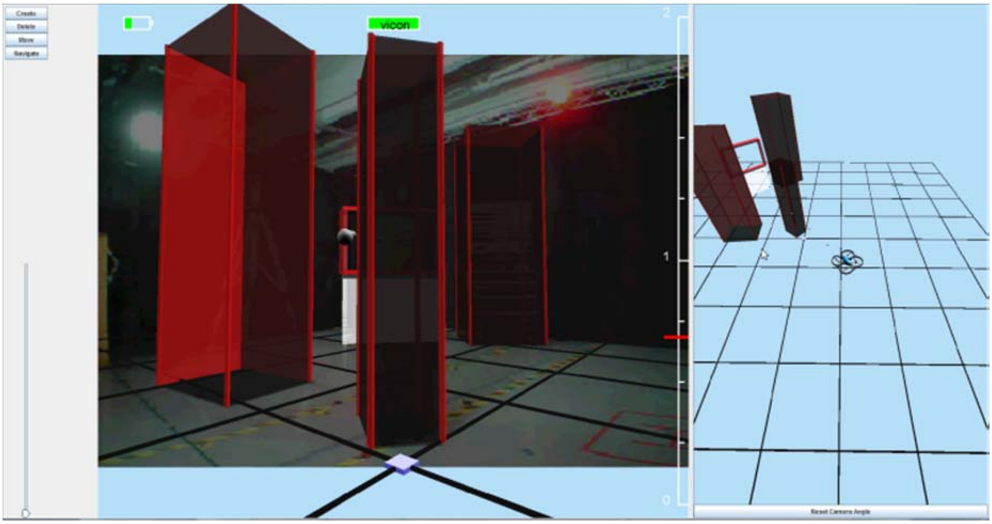



Figure 38 Human-MAV Team Navigation Interface. A common frame of reference is used for both the live video perspective (left) and the 3D world model (right).

The left view may seem similar to the normal camera view that might be presented to a teleoperator, but there is a significant difference. This video is embedded in a 3D world model. This provides several advantages. First, it provides a common frame of reference for interaction. This is critical to enabling joint activity between the human and the machine. This allows the creation and manipulation of objects in 3D space in a manner compatible to both the human and machine. Second, the field of view can extend beyond the limits of the camera. Notice how some of the objects project outside the video in Figure 38. The operator is also not limited by the bounds of the video for object creation, which can be very useful in tight spaces.

The right view can provide an overhead view common in many systems, but it is not limited to this perspective. The viewpoint is navigable to an infinite number of possible perspectives to suit the needs of the operator.

To meet the directability requirements of Figure 37, the operator interacts with the system by an intuitive click-and-drag method common to many 3D modeling tools. The mathematics behind the interface our presented in our previous work with ground vehicles (Carff, Johnson, El-Sheikh, & Pratt, 2009). The operator can create walls and obstacles to limit where the vehicle can go. The operator can also create doors and windows to indicate where the vehicle can go. Figure 39 shows some example objects. Objects can be stacked to create complex structures. These simple tools allow the operator to effectively model the environment. Our current system provides no autonomous perception of objects, but by designing it as we have, we can incorporate such input in the future. The main difference would be that our interface ensures the operator can not only see the results of the autonomous perception, but also have the ability to correct, modify and add to those results as a team member.

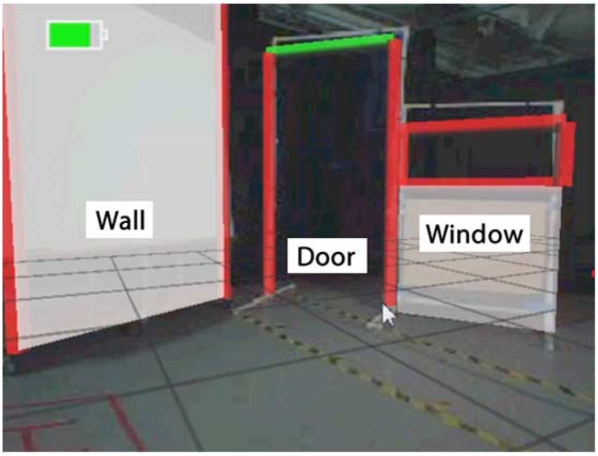



Figure 39 Examples of objects created by an operator.

Paths are generated autonomously by clicking on a location or by choosing an object, such as a door or window. The path is displayed for the operator to see prior to execution, as shown in Figure 40. They can be modified as necessary using a variety of ways provided by our interface to influence the path of the vehicle. Multiple paths can be combined to create complex maneuvers.

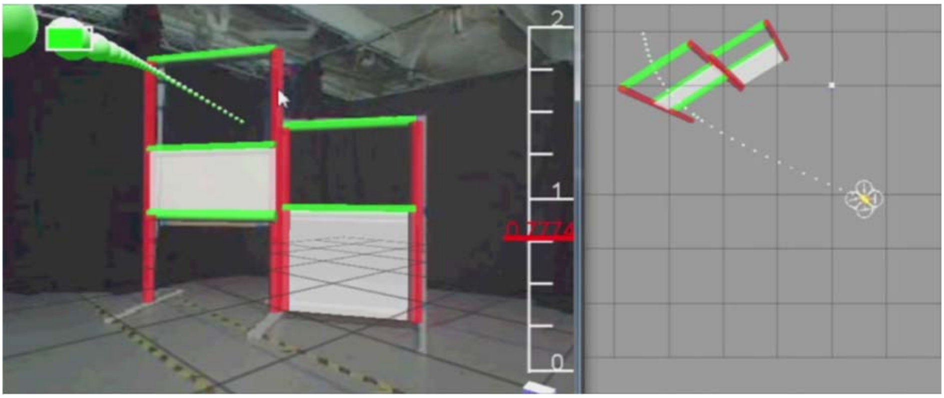



Figure 40 Autonomously generated path (green balls) displayed in both the live video and the 3D world model.

## 10.6 System Features

Our system allows collaboration throughout the navigation task including during perception of obstacles and entryways, during decision making about path selection and during judgment about standoff ranges. As such, our unique approach affords the operator the ability to do things that are not possible with conventional video and overhead map interfaces.

### *10.6.1 Onboard sensing error observation and correction*

By providing a common frame of reference we can make the internal status of the vehicle apparent to the operator. Figure 41 shows a typical situation in which the onboard sensing has accumulated some error over time. This error is manifested as an offset between the virtual objects and their real world counterparts in the live video. This provides a very intuitive way for the operator to understand how well the vehicle is doing. Not only can the operator see the problem (observability), but we also provide a mechanism to fix it (directability), directly addressing two key issues in Figure 37. The operator can simply click-and-drag the virtual object to the correct location and this will update the vehicle's localization solution.

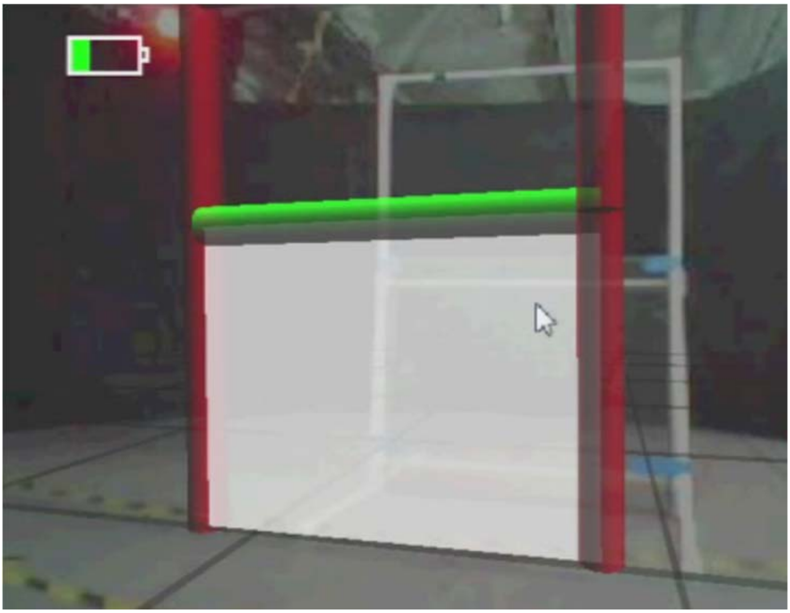


Figure 41 Onboard sensing error visualized through our interface. The difference between the real window and the virtual window is an accurate measure of the MAV's onboard sensing error due to drift in the MAV's position estimate. The operator can click-and-drag the virtual window to correct this error for the robot.

### *10.6.2 Preview*

We can provide the operator a virtual preview of the flight before committing to it (predictability). Cooper notes the importance of predictability and the advantages it provides a system (Cooper, 2007). Once a path is chosen, the operator simply requests a preview and a virtual drone will fly the selected path, as shown in Figure 42. The virtual drone is visible on both the live video and the 3D world model, allowing the operator to have multiple perspectives of the flight. By displaying a full size model, the operator can see the flight in context of the vehicle size in order to better judge obstacle clearance, meeting another requirement in Figure 37. The operator can try out alternative solutions before committing to the best one for execution.

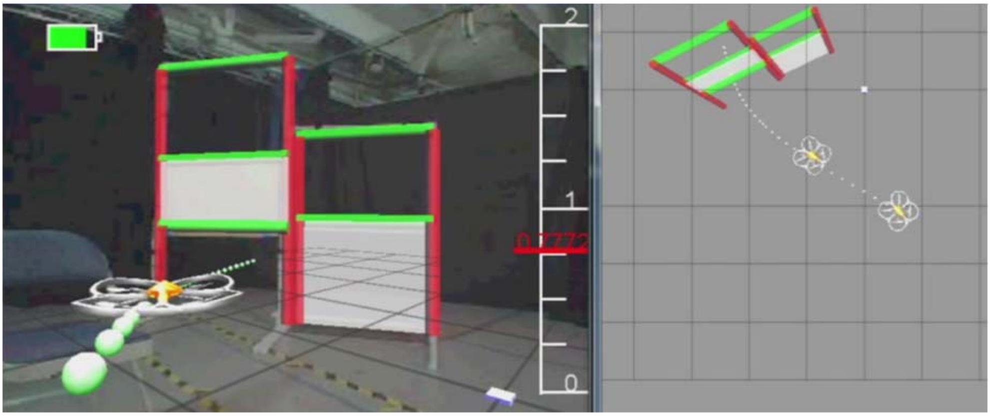



Figure 42 A preview of a flight displayed in both the live camera view and the 3D world model view. Prior to execution of the flight path, the operator can request a preview to see the path in the context of the vehicle size. The virtual MAV is a prediction about MAV behavior during execution.

### *10.6.3 Third Person View*

Another ability of our system is a third person perspective that allows the operator to view the vehicle from behind; enhancing situation awareness about the proximity to nearby obstacles outside the field of view of the onboard camera. The value of this type of approach has been shown to improve navigation performance (Michaud et al., 2010; Nielsen et al., 2007). We use historical images and a virtual MAV to enable the operator to see the vehicle from a third person perspective. For example, it would be difficult to fly exactly to the corner of the wall in Figure 43 since the corner would be outside the field of view before the vehicle was in position. It would also be difficult to judge proximity to the wall, particularly once it leaves the field of view. Our third person view lets the operator accurately judge proximity and maintain a highly accurate position relative to the corner even when outside of the normal camera field of view. It is important to note that the common reference frame makes the multiple perspectives useful, instead of it being an additional burden to the operator.

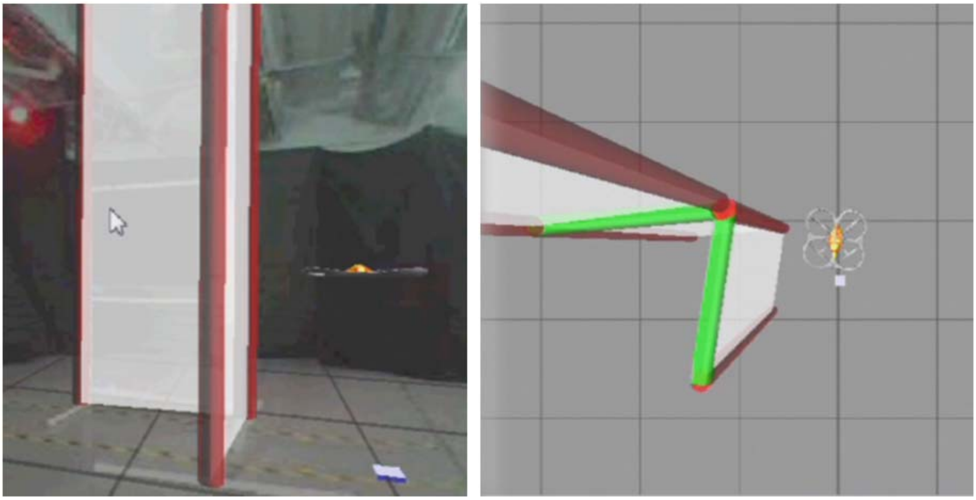


Figure 43 Example of third person view. The virtual MAV in both views represents the actual position of the real MAV. The left view lets the operator watch the MAV from behind. The right view is currently oriented to let the operator watch the vehicle from above.

### *10.6.4 Support for Operator Preference*

Engineers love to design optimal solutions, however, human operators rarely agree about what is optimal. Should it be the fastest route, the safest route, or something else? Our system allows human adjustment to tune system behavior in a manner that is compatible with the operator's personal assessment of optimal. For example, we provide an adjustable buffer zone, shown in Figure 44, which can be used by the operator to vary the standoff range from obstacles during planning and execution. This buffer zone could be used to provide additional clearance around a fragile object or it could be used to provide a safety buffer for a vehicle that is experiencing navigational error. This type of interaction can help improve operator acceptance of the system, by calibrating system performance to the operator's comfort level.

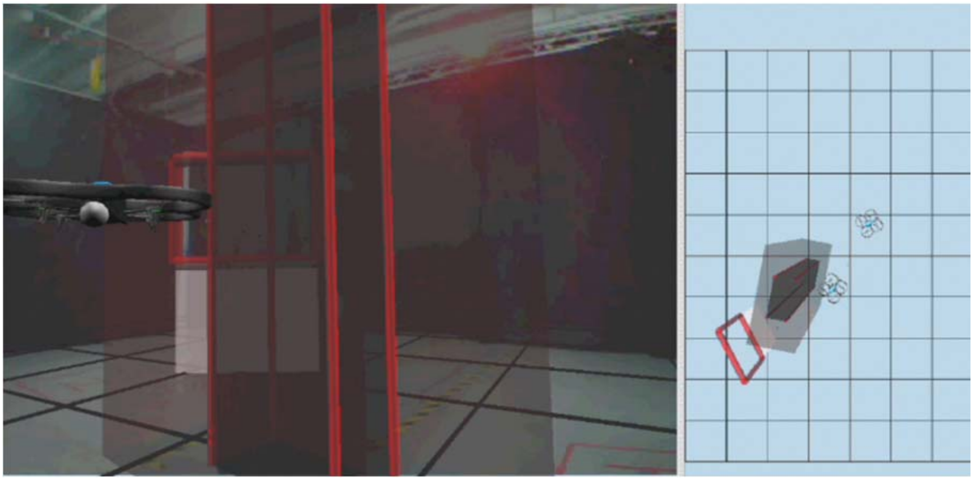


Figure 44 Example of adjustable buffer zone around obstacle

### *10.6.5 Enabling Creative Solutions*

Since our interface treats the operator as an equal partner in the navigation solution, we do not limit the operator to solutions generated by autonomous algorithms. The operator has the freedom to apply their creativity to the solution. Some examples that permit creativity include how to model the environment, simplification of maneuvers and flexibility with vehicle orientation.

There is often little need to accurately model everything in the environment in order to achieve a goal. Human judgment about relevance can simplify the problem, making it only as complex as needed. Consider our cluttered environment in Figure 34. Do we need to model everything in view as shown in Figure 45? This is probably not the case for most situations. One could just model the nearest obstacles to the flight path of interest, as shown in Figure 46. Instead of modeling obstacles, an alternative approach is to model the solution by using doors and windows as "gateways" connecting zones of safe passage, as shown in Figure 47. This type of interaction can result in a more robust system, by leveraging the creativity of the operator to overcome circumstances unforeseen by the system's designers.

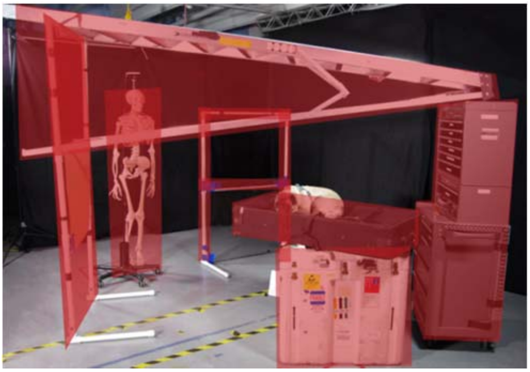


Figure 45 Example of unnecessary modeling of all objects.

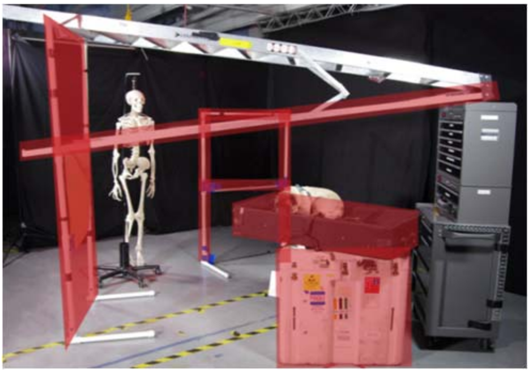


Figure 46 Example of modeling only the objects nearest the intended path.

Figure 47 Example of modeling "gateways" of safe passage using doors and windows.

Some maneuvers are more challenging than others. Our interface provides the opportunity to reduce the complexity of some maneuvers, particularly in confined spaces. Consider the task of flying into a narrow corridor, observing something on a wall and exiting the corridor. Turning around is a very challenging teleoperation task, since the operator has a limited field of view and tight spaces offer limited visual cues. Our interface affords a creative solution to the challenge. The operator can rotate the vehicle prior to entering the space, since our alternative perspectives, such as the one shown in Figure 48, allow navigation without requiring the use of the camera view. With this, the maneuver is reduced to a basic lateral translation into and out of the space, which is a much easier maneuver than a rotation while inside the confined space.

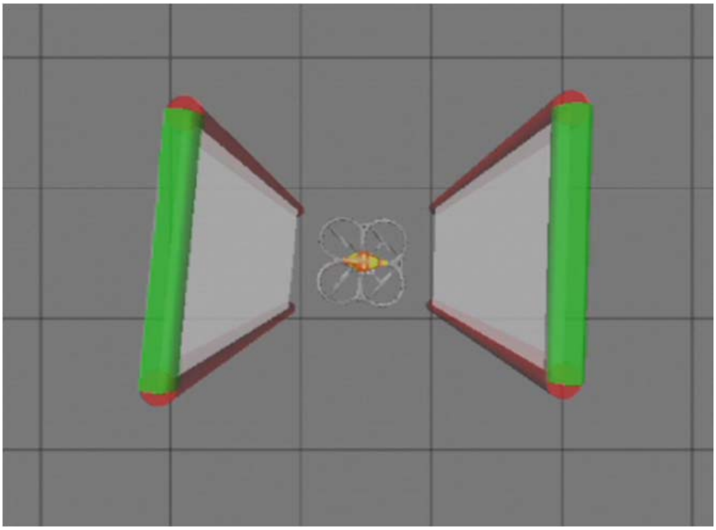


Figure 48 Simplified navigation in confined spaces. By using the overhead view, the operator is not reliant on the forward facing camera view to navigate, allowing a lateral translation into the confined space rather than a more difficult rotation while inside the confined space.

Our interface affords some unique possibilities by not having to rely on the camera view at all times. It enables the potential for obstacle avoidance even when the vehicle is not oriented toward the direction of motion. This allows the vehicle to keep the camera on a point of interest while still avoiding previously annotated obstacles.

These are a few of the creative solutions possible with our unique approach.

## 10.7 Results

With our human-MAV team navigation system we were able to successfully navigate through a variety of obstacles and negotiate tight spaces. The system is designed to be used online during the flight. It takes approximately 3-5 seconds to mark up a typical obstacle. Occasionally maneuvering is required to see all the relevant objects and it typically takes 15-30 seconds to mark up a scene. Once marked up, our typical flight took approximately 15-30 seconds to navigate the obstacles and reach the goal. While our system basically doubles the flight time, one must consider that the resulting flight is a single continuous movement through the environment. Normal teleoperation would typically involve some pausing and orientation during the traversal, resulting in a slower flight time. Future work will involve experimental evaluation of these rough estimates and verification of the performance measures of the system.

## 10.8 Conclusion

This project has demonstrated the unique type of human-machine system that can be developed when interdependence is given proper consideration in the design process. The operator could observe the internal state of the vehicle by the relative location of graphical objects. The operator could predict the resulting behavior prior to execution by a displayed path or even a virtual "fly through." Directability was supported in a variety of ways from goal specification, to waypoint modification, to obstacle correction, to state estimation adjustment. These are just a few of the ways interdependent relationships were supported to provide a lot of flexibility. We feel our interface provides a truly collaborative experience, allowing the human to participate in sensing, perception, planning and judgment. The project also highlights new capabilities, impossible with most current systems, but made possible by taking a coactive design perspective on the problem. This includes things like flying from a third person view and enabling safe flight at angles orthogonal to the camera view. It is important to note that the flexibility is not a particular feature, such as allowing for graphic overlays. Flexibility is the additional options that a feature affords. The features in this UAV system demonstrate the type of flexibility a coactive system can provide.

***We are stuck with technology***
***when what we really want is just stuff that works.***

**– Douglas Adams**

# 11 Applying Coactive Design in the DARPA Virtual Robotics Challenge[22]

Coactive Design was developed specifically to address the increasingly sophisticated roles that people and robots play as the use of robots expands into new, complex domains. DARPA recently hosted a competition, called the DARPA Robotics Challenge (DRC) that is an example of the type of new and complex domain to which we refer. We participated in the competition as part of the IHMC team. The competition afforded a way to evaluate the Coactive Design approach against traditional approaches. The coactive system we developed was quite different from any of the other twenty-six entries in the competition. It also afforded an opportunity to evaluate the ability of the Coactive Design approach to imbue resilience. The results of the first phase of the competition, the Virtual Robotics Challenge (VRC), are presented as an exemplar of large scale implementation of Coactive Design.

We used the Coactive Design approach as the basis for our overall system design for the DRC. We made extensive use of the Coactive Design method and the IA Tables. We will use this domain as an example of how the Coactive Design method can be operationalized. This includes demonstrating how analyzing interdependence helps enumerate the potential design options. We will also show how it can help identify constraints and requirements. We provide specific examples of how our method led to a specification sufficiently detailed to guide implementation. Finally, we show how the analysis also aides in understanding the impact of design choices.

## 11.1 The DARPA Robotics Challenge

The DRC was created to spur development of advanced robots that can assist humans in mitigating and recovering from future natural and man-made disasters. The VRC was the first phase of the DRC. It was a software competition carried out in a virtual environment that looked like an obstacle course set in a suburban area. The competition involved remotely operating a simulated Atlas humanoid robot.

[22] This chapter is adapted from (Johnson et al., 2014)

The robot has 28 actuated degrees of freedom, a stereo camera, and a laser range finder. There were three tasks to complete, as shown in Figure 49. The first was navigating complex terrain that included mud, hills, and debris. The second task was picking up a hose, attaching it to a spigot, and turning a valve. The third task required entering a vehicle, driving on a road with turns and obstacles, and getting out of the vehicle. While some parts of these tasks have been demonstrated by various researchers, the scope and breath of these challenges raises the bar for humanoid capabilities.


Figure 49 DARPA Virtual Robotics Challenge tasks. They included walking through mud, walking over hills, walking through debris, entering a vehicle, driving along a road, avoiding obstacles while driving, exiting a vehicle, picking up a hose, attaching the hose to a spigot, and turning on a valve.

The competition took place over a grueling 56-hour period. Each team needed to complete five examples of each of the three tasks. The five examples were created by DARPA, and information about the examples was withheld from the teams prior to the competition. Each example had some variability, such as the position of objects, the color of objects, the location of obstacles, and even damping values on the valve and the mud. Teams were allotted 30 minutes for each attempt, which meant there was a possibility of up to 7.5 hours of operation time. The simulations ran “in the cloud,” and a minimum 500ms of network latency was imposed on all teams. Teams were ranked based on the number of tasks successfully completed (points), the time to complete the tasks, and the amount of bandwidth used. After initial entries from 126[23] potential competitors, 26 teams from eight countries qualified to compete in the VRC. The top nine teams were listed in the final results of the competition, as shown in Table 6.

Table 6 DARPA Virtual Robotics Challenge (VRC) Results. Score equates to number of tasks successfully completed. Falls is the number of times the robot fell. Banked up bits is a measure of bandwidth usage (higher is better). Banked time is a measure of task completion speed (higher is better).

| Rank | Team | Score | Falls | Banked Up Bits | Banked Time |
|---|---|---|---|---|---|
| 1 | Institute for Human and Machine Cognition (IHMC) | 52 | 12 | 95 | 13,813 |
| 2 | Worcester Polytechnic Institute (WPI) | 39 | 12 | 99 | 13,545 |
| 3 | Massachusetts Institute of Technology (MIT) | 34 | 20 | 77 | 6,829 |
| 4 | TRACLabs | 30 | 19 | 98 | 16,171 |
| 5 | NASA JPL/UCSB/Caltech | 29 | 22 | 98 | 13,209 |
| 6 | TORC/TU Darmstadt/Virginia Tech | 27 | 25 | 85 | 13,421 |
| 7 | Team K (Japan) | 25 | 16 | 84 | 10,442 |
| 8 | TROOPER (Lockheed Martin/University of Pennsylvania/Rensselaer Polytechnic Institute | 24 | 27 | 76 | 13,927 |
| 9 | Case Western University | 23 | 29 | 81 | 10,951 |

## 11.2 Applying Coactive Design in the VRC

The first way Coactive Design impacted our design decisions in the VRC was to shift our engineering focus from developing autonomy to developing a human-machine team. This was important, especially given that DARPA introduced bandwidth limitations with the stated purpose of encouraging autonomous solutions. We were willing to accept the bandwidth penalty (i.e., coordination cost) to gain the benefits of teamwork. For our purposes, we considered the team to consist of the Atlas humanoid robot and an operator. The Atlas robot operated within a Gazebo physics simulator (Koenig & Howard, 2004). The human operator was remote and could sense the world only through the data presented on the operator interface. We will focus on one subtask from the VRC to demonstrate how Coactive Design was operationalized in our VRC entry. Following the Coactive Design method from Figure 18, an IA table for the subtask of picking up the hose was constructed in accordance with the chapter 8.1 on the identification process. The resulting IA table is shown in Figure 50.

[23] http://www.darpa.mil/NewsEvents/Releases/2013/06/27.aspx (Accessed 2013-11-19 0900CST).

| Tasks | Hierarchical Sub-tasks | Required Capacities | Team Member Role Alternatives | | | |
|---|---|---|---|---|---|---|
| | | | Alternative 1 | | Alternative 2 | |
| | | | Performer | Supporting Team Members | Performer | Supporting Team Members |
| | | | Robot | Human | Human | Robot |
| Find the hose and pick it up | Locate the hose | Sense the hose | | | | |
| | | Recognize the hose | | | | |
| | Walk up to the hose | Understand the arm workspace | | | | |
| | | Know where to stand for picking up the hose | | | | |
| | | Specify the footsteps to position the robot | | | | |
| | | Execute the footsteps | | | | |
| | Grasp the hose | Know how to grasp the hose | | | | |
| | | Position the hand for grasping | | | | |
| | | Execute the grasp | | | | |
| | | Verify the grasp is good | | | | |
| | Lift the hose | Raise the hose for carrying | | | | |
| | | Abort if grasp is improper | | | | |

Figure 50 Interdependence analysis for VRC subtask of picking up the hose.

### *11.2.1 Identification Process*

The task decomposition and required capacities were determined by careful consideration of the taskwork. We had only two team role alternatives for the VRC, as seen in Figure 50. The capacity assessment is from a particular point in development. It is important to realize that the IA table needs to evolve with the design. As an example, the human is remote and has no way to sense the hose without some interface providing sensor data. Figure 50 was generated after a basic operator interface was developed to provide video to the operator, so Figure 50 shows that the human has the capacity. From this we can generate a set of interdependence relationships and their associated observability, predictability and directability (OPD) requirements, as shown in Figure 51.

The identification process does not dictate what needs to be done. Instead it helps identify the available options. Sometimes there is only one option. Footstep execution is such a case, since the operator can contribute little to dynamic balancing when there is 500 ms of network latency. This indicates a critical path in the design and resulted in our focusing a large amount of effort to ensure reliability of our walking algorithm. Other times there are multiple options, but one is the clear choice. Recognizing the hose is an easy task for the operator. We could have expended resources developing an autonomous hose recognition algorithm, but it would never be as reliable as the human. Occasionally the choice is not clear. For example, either the robot or the operator could position the hand for grasping, but neither was 100% reliable. In these cases it is often beneficial to support both approaches, which adds flexibility to the system. If the robot fails to position the hand correctly, then the operator can also try. There are actually more than two options in positioning hands for grasping. Besides allowing either the robot or the human to attempt the action, there are the additional potential interdependence relationships. These are indicated by the yellow capacity to support for each team member. The yellow also means they are optional alternatives, not required ones. For example, the human could provide updated information about the position of the hose in order to improve the robot positioning of the hand or the interface could provide more intuitive mechanisms for positioning control of the six degrees of freedom in each arm.

The end result of the identification process is a set of interdependence relationships and associated OPD requirements, such as those shown in Figure 51. These are used as criteria to determine if a particular mechanism we developed to support these requirements are sufficient as well as in the process of evaluating the effects of a change.

| Required Capacities | Team Member Role Alternatives 1 | | 2 | | Interpretation and OPD Requirements |
|---|---|---|---|---|---|
| | R | H | H | R | |
| Sense the hose | | | | | Either can sense the hose independently. |
| Recognize the hose | | | | | Only the operator can recognize the hose, so there is an observability requirement to make the position and orientation known to the robot. |
| Understand the arm workspace | | | | | The operator will need assistance understanding the arm workspace, so valid positions and limitations need to be made observable. It should also be predictable when the position cannot be achieved. |
| Know where to stand for picking up the hose | | | | | The robot can accurately perform this task but cannot account for additional context such as obstacles, therefore the robot's position must be directable by the human. The human cannot accurately and repeatable select the position without some directability provided by the interface. |
| Specify the footsteps to position the robot | | | | | The robot cannot reliably perform this because it lacks obstacle detection. The human must make obstacles observable to the robot. The human cannot reliably perform this because it is difficult to evaluate the terrain. The interface must provide observability into the suitability of the terrain. |
| Execute the footsteps | | | | | The robot is the only viable option. |
| Know how to grasp the hose | | | | | The operator is the only viable option and may not be reliable. |
| Position the hand for grasping | | | | | Either can position the hand. Occasionally the robot fails to achieve the desired position and the operator needs observability to know when this has occurred. The human can control each joint, but this is tedious and could use some assistance achieving feasible end effector positions reliably. |
| Execute the grasp | | | | | Either can execute the grasp. |
| Verify the grasp is good | | | | | Only the operator can verify the grasp. This observability issue requires excellent awareness of the hand position and the object being grasped. |
| Raise the hose for carrying | | | | | Either can raise the hose for carrying. |
| Abort if grasp is improper | | | | | Only the human can detect an improper grasp. This will require directability to release and reposition the hand and/or directability to alter the raising of the hose to be more cautious in order to avoid dropping the hose. |

Figure 51 VRC hose task interdependence relationships and OPD requirements.

## *11.2.2 Selection and Implementation Process*

Verification of the grasp is a critical task that requires excellent awareness of the hand position and the object being grasped. Spatial awareness is a recurring issue in remote operation and was essential to all of the VRC tasks. As such, we developed an interface (shown in Figure 52) based on 3D world model that is updated by state estimation from the robot. The left side provides a live video stream embedded in a 3D world model fixed to the first-person perspective. The right side is a navigable third-person view of the same 3D world model, allowing the operator to take any perspective. The third-person view proved invaluable with the operator spending the majority of time focused on this perspective. Without it, there would be no way to verify the grasp, since the hand covers the hose when grasping from a table. During the five hose tasks of the VRC, the operator made an average of 34 perspective changes in the third-person view, indicating how often this view was relied upon for situation awareness. Additionally, of the commands issued through the 2 views, 87 percent were issued through the third-person view.

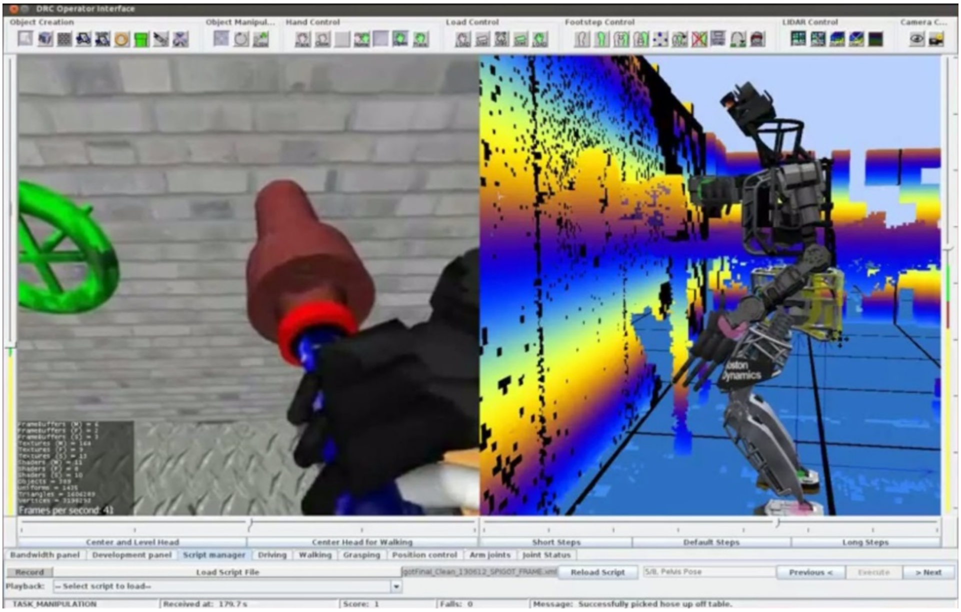



Figure 52 IHMC's VRC operator interface. The foundation of the interface is a 3D world model that is updated by state estimation from the robot. The left side provides a live video stream embedded in a 3D world model fixed to the first-person perspective. The right side is a navigable third-person view of the same 3D world model, allowing the operator to take any perspective.

Our robot did not have the ability to recognize the hose, so in order for the robot to participate in "walking up to the hose" or "grasping the hose," it needed to be made aware of the position and orientation of the hose. Since we developed graphical 3D world model, we decided manipulables would be a good way to meet this requirement. Manipulables are visual representation of things that we needed to communicate about, such as the hose, which can be placed into the 3D world model or on the live video and be repositioned as desired using simple click-and-drag techniques common in computer-aided design (CAD) programs. Figure 53 shows the hose manipulable. Since our model is driven by the state estimation of the robot, the onboard error is visually represented by drift of the manipulable (i.e., observability into the robot's state estimation). The use of manipulables provides a means to correct the error by simply dragging the manipulable to align with the sensor data (i.e., directability into the robot's state estimation). We also used this manipulable as a reference for where to stand and as input to planning algorithms when generating footsteps for the "walking up to the hose" task. Manipulables proved so valuable that they were consistently used in all five hose task runs of the VRC. More telling of their value is the thirty-four adjustments made after their initial placement to correct for deviations. Failure to account for these deviations could easily have caused an error or even prevented us from successfully completing the task.

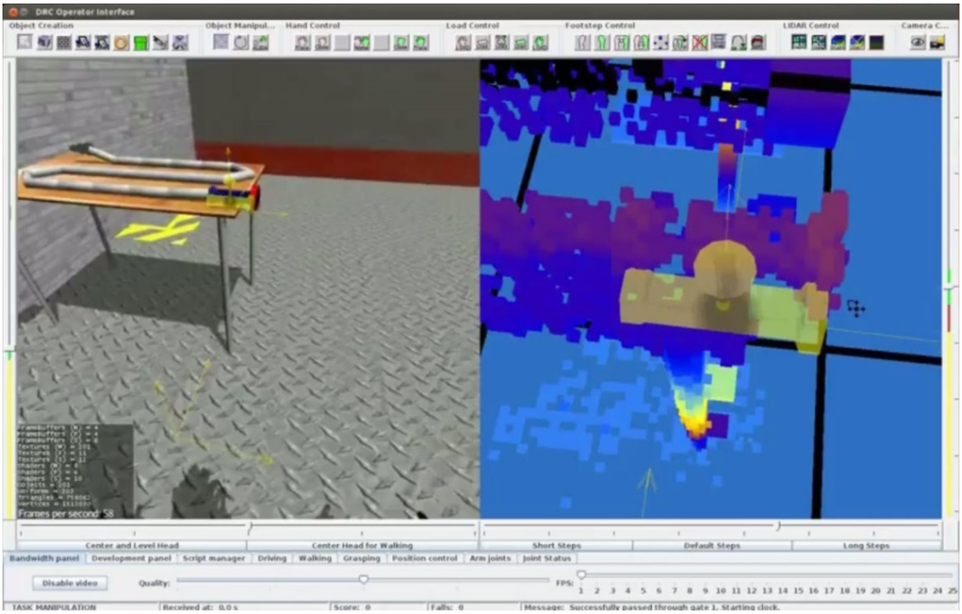



Figure 53 Hose manipulable (yellow) shown as a virtual object on both the live video (left side) and the 3D world model (right side).

A key aspect of the human being able to participate in the task was assisting the human in understanding the workspace of the robot's six degrees of freedom arm. Inverse kinematics is the mathematical way the robot solves this, so we made virtual arms that displayed the inverse kinematic solution to the operator prior to any execution (i.e., predictability). This made arm limitations observable to the operator. It also provided predictability by changing the color from green (valid) to red (invalid), as shown in Figure 54. This ensured the operator was aware that the arm could not achieve invalid solutions. The virtual arms were also manipulables that provided a much easier way to position the hands (i.e., directability) than trying to control all six degrees of freedom individually. In fact, the virtual arm manipulable was used in 99% of all arm commands during the five hose task runs of the VRC. Even though the virtual arms were extremely effective, we maintained support for other interdependence alternatives. This was important, because without supporting the one percent by maintaining support for joint level control, we would have failed three out of the five runs that required it.

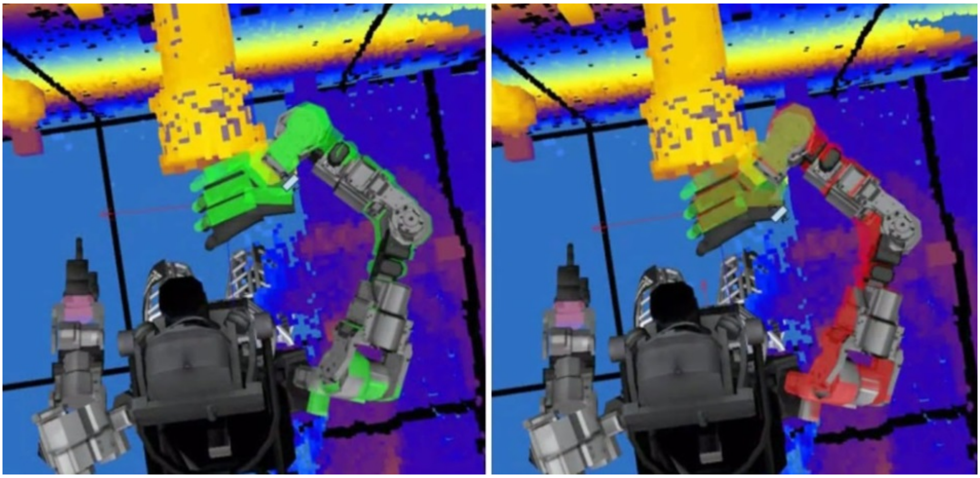


Figure 54 Inverse kinematics visuals inform the operator of valid (left side - green) and invalid (right side - red) solutions.

In support of the requirement for easier and more accurate hand positioning for grasping, we developed a graphical element representing the valid grasp region of the hand. Robot hands are not as compliant as human hands and only similar in a very limited number of ways. One limitation is reflected in the effective grasp region. We made this region visible, as shown in Figure 55. This location is not a joint, but using the manipulable allowed the operator to control the hand around this point if desired. By enabling this, the operator could to do things such as rotating the hand around the object being grasped in order to grasp from a different direction. The feature was used in both the hose task and during car entry and exit where the robot needed to grasp the roll cage of the car.

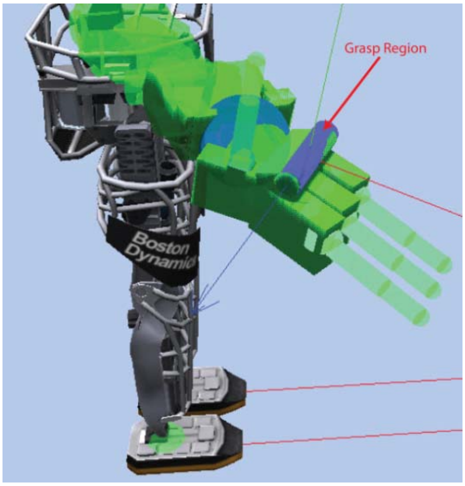



Figure 55 Grasp region visual element used to assist with easier control of hand position.

These are a few examples of mechanisms we developed to meet OPD requirements in support of interdependence relationships. While a creative designer could have developed these without Coactive Design, our approach provides both a repeatable methodology, a reasonable set of evaluation criteria, and a way to evaluate change.

### *11.2.3 Evaluation of change*

Evaluation is an important part of the design process. It is important to not only validate that mechanisms selected and implemented meet requirements, but to assess their impact on the rest of the system. Some of our decisions had a positive impact across multiple tasks, such as the use of a third-person perspective. Other choices only impacted the specific requirement they were targeting, such as the grasp region. However, other choices had the potential to have a negative impact by impeding OPD requirements or altering interdependence relationships. An excellent example from our VRC work is the use of scripting.

Given the fairly limited scope of the hose task in Figure 50, it is conceivable that the entire process could be automated. Our team attempted to automate just the grasping and lifting of the hose portion. Our approach was to generate a script, or sequence of actions, that recorded the successful execution of the task. The script could then be played back in order to automate the process. This choice of implementation for the automation process eliminated any potential support for interdependence. The IA in Figure 50 provides indication of what might (and did) go wrong. First, there was no capacity for the robot to verify its own grasp. By automating the process, we removed the opportunity for the operator to verify that things were going well. After many frustrating failures, the evaluation determined that this approach was too brittle, so we enabled step-by-step playback of the script with supporting visuals to make the upcoming action observable and predictable. This afforded the operator a chance to verify the grasp. Evaluation of this approach was also deemed insufficient, because failure meant aborting the process and rescripting. The main issue was the robot's reliability in positioning the hand for grasping was less than 100 percent (yellow). The solution was to include directability, allowing the operator not just to see the upcoming action but also to modify it if necessary, or replay it, or even skip it if desired.

The coactive solution to scripting proved a flexible and resilient one. During the five hose tasks of the VRC an average of 10 scripts were used per run. Only 50 percent of these were run without intervention. We averaged nine pauses in script behavior to verify performance and seven operator corrections to scripted actions per run. Even with operator intervention, 8 of the 50 scripts failed to accomplish their purpose. Due to the flexibility in our system to retry, make adjustments, and use different approaches, we were successful in recovering from all eight failures.

In the end, the IA provided insight into how design decisions, such as automating a task, might impact the overall system. Our resulting solution allows for autonomous behavior but with appropriate support for interdependence, i.e., the human can participate in the activity in a collaborative manner.

## 11.3 Advantages of Using the Coactive Design Approach in the VRC

Our approach to the hose task illustrates only a few of many ways we designed and built the system to support interdependence. Designing for interdependence provided our team several advantages.

The first advantage is *flexibility*. We could perform the same task in many different ways. Our approach to scripting the hose task is an example of how including support for interdependence can provide flexibility. Flexibility was important in other tasks as well. For example, different walking challenges (e.g., mud, hills, debris, and flat open ground) made some approaches more attractive than others in a given instance. The operator was relatively unburdened in handling walks over flat, open ground because the system could be allowed to work more or less "autonomously." However, when the robot was walking over more difficult terrain, operators could seamlessly increase their involvement in the task with no need for a major mode switch. Flexibility was also important because we were not privy to the specifics of each task a priori and had to deal with uncertainty.

*Resilience* was a second advantage. If we encountered an unexpected problem—whether it was related to the unforeseen challenges of the task or difficulty in achieving the expected system response—our flexibility allowed us to try different approaches. Benjamin Franklin is often cited as saying, “If you fail to plan, you plan to fail.” In robotics, if you do *not* plan to fail, you are failing to plan. Uncertainty and unexpected events are part of robotics and designing resilience into a system is how to address this reality. A good example of this occurred during one of our driving tasks. The car went off the edge of the road during a sharp turn and required us to back it up to avoid a building. We had an “autonomous” method of switching the car into reverse, but it failed because the robot had been jostled as the vehicle left the road and was no longer in the correct position. The observability we had built into the system allowed the operator to correctly assess the problem and the additional directability options we provided enabled engagement of reverse by alternative means. The score in Table 6 is a measure of completion. We successfully completed 52 out of a possible 60 points (86.7 percent). This is not because we performed flawlessly. We fell 12 times and encountered numerous unanticipated circumstances. Our score reflects our system’s resilience to recover from these challenges and adapt to overcome them.

A final advantage we will highlight here is *development efficiency*. The VRC competition was a complex design challenge with an unbelievably short time period. Teams had nine months to prepare. This forced a lot of tradeoffs in our design considerations. The Coactive Design approach provided an excellent way to perform a cost-benefit analysis of design choices. An example from the hose task was the decision not to pursue autonomous hose recognition. Similarly, “autonomous” obstacle avoidance for walking relies heavily on perception and was likely to be frail in the end. Our team saved a lot of time by not investing in complex perception and planning while instead focusing our resources on enabling the human to be an effective teammate. By strategically ruling out the 100 percent solution (i.e., full autonomy or full teleoperation) we could deftly avoid the hardest problems. Our Coactive Design approach led to a system that could exploit synergy between the machine and the human—in essence, allowing us to work as a team.

## 11.4 Conclusion

We demonstrated how Coactive Design was operationalized by the IHMC team during the DARRA VRC. *Our success was not based on flawless performance, but on resilience in the face of uncertainty and misfortune and surprise.* We propose that it is through understanding and modeling interdependence in a human-machine system that Coactive Design can play a role in enabling robots to fulfill their envisioned role as teammates.

***We shall not cease from exploration***
***And the end of all our exploring***
***Will be to arrive where we started***
***And know the place for the first time.***

***– T. S. Eliot***

# 12 Conclusion

The journey of this research has been challenging and enlightening. Looking back at our starting point, the original kernel of inquiry remains, but the surrounding context looks very different. The T. S. Eliot quote at the head of this chapter captures this sentiment. While the work itself is of value, it is the new understanding that we carry forward into future work that will have a lasting impact.

Here, it is appropriate to return to the primary question addressed in this thesis: How does one design a resilient system? We have argued that resilience is achieved through designing for flexibility. Specifically, by providing alternative ways to recognize and handle unexpected situations. Flexibility of a team is enabled by a teamwork infrastructure that supports a variety of interdependence relations. These relationships allow members of a human-machine team to recognize problems and adapt, thus benefitting resilience. Coactive Design breaks with traditional approaches by focusing on effective management of the interdependencies instead of focusing on autonomy.

Our research aim was to provide a design method for identifying and exploiting interdependence in human-machine systems, in order to provide ways to recognize problems and create alternatives to address them. The method we developed is called the Coactive Design method. The method is supported by a tool called the Interdependence Analysis Table which helps identifying and exploiting interdependence in human-machine systems.

This thesis centers around two primary research claims. The first is: Interdependence is an effective basis for a design and analysis model of human-machine systems. The detailed discussion of Chapter 6 shows interdependence to be the key for understanding the complexity of human-machine systems. The interdependence analysis provides the path by which we can add not only capability, but also the kind of flexibility that enables a system to be resilient. The second claim of this thesis is: Resilience in human-machine systems benefits from a teamwork infrastructure designed to exploit interdependence. Evidence for the need of such a structure is provided by the BW4T experiment in which increasing autonomy *without* providing a teamwork infrastructure resulted in degraded performance. The benefits of including an infrastructure that supports interdependence are demonstrated by the UAV and DRC systems that proved themselves capable of surviving and successfully completing complex operations in scenarios with unexpected events. These case studies demonstrate that the flexibility of designs based on interdependence indeed leads to resilience.

Before concluding our journey, we will summarize the results we have achieved.

## 12.1 Results

Coactive Design makes five major contributions: 1) a new perspective, 2) a richer understanding of interdependence, 3) a new system model, 4) a new design method, 5) and a new tool to assist the designer in system design and analysis.

The first contribution of Coactive Design is a change in focus (Chapter 5). Focusing on interdependence is a clear break from the autonomy-centered perspectives that dominate current research. Coactive Design is focused on systems where the human and machine are engaged in teamwork. Besides implying that more than one party is involved, the term "coactive" is meant to convey the *type* of involvement. Consider an example of playing the same sheet of music solo versus as a duet. Although the music is the same, the processes involved are very different (Clark, 1996). The difference is that the process of a duet requires ways to support the interdependence among the players. This is a drastic shift for many autonomous robots, most of which were designed to do things as independently as possible. The term "coactive design" is about designing in a way that enables effective teamwork through support for interdependence.

The second major contribution of this work is a definition of interdependence and an understanding of the design implications of this definition (Chapter 6). The central role of interdependence demands a rich understanding of interdependence itself. In his seminal book, James D. Thompson (1967) recognized the importance of interdependence in organizational design, just as we are proposing its importance in human-machine systems. The correlation is made clear by Thompson's description of an organization as an "open-system, indeterminate and faced with uncertainty" (p. 13). He also noted that there was a lack of understanding about interdependence — something still true today. Understanding the nature of the interdependence between team members provides insight into the kinds of coordination that will be required of them. Indeed, we assert that coordination mechanisms in skilled teams arise largely because of such interdependencies. For this reason, understanding interdependence is an important requirement in designing systems that will work jointly with people. This thesis argues that managing interdependencies is the mechanism by which we achieve the higher level concepts of coordination, collaboration and teamwork.

The third major contribution is a new system model for human-machine system design (Chapter 7). We have already referred to the need to "manage" interdependencies and to "support" interdependent relationships — this chapter begins to describe how we think this may be done. Our system model highlights three key team capabilities, over and above task capabilities, that are needed for effective human-machine collaboration: observability, predictability and directability. For team members, these three capabilities enable resilience, allowing them to "recognize and adapt to handle unanticipated perturbations" (Woods & Hollnagel, 2006, p. 22). From a designer's perspective, observability, predictability, and directability are important because they provide guidance on how to identify design requirements. By determining how these capabilities must be supported in order to be capable of understanding and influencing team members, designers can create a specification. This design stance necessarily shapes not only the "user interface" for the human but also the implementation of a robot's autonomous capabilities. The shaping process is provided by the three team capabilities in our system model which capture three of the key elements required for effective teamwork.

The Coactive Design *method* is the fourth major contribution (Chapter 8). It is a method for designers building highly interdependent systems. It provides the first step by step procedure for designing interdependent systems, based on the perspective provided by Chapter 5, the understanding provided by Chapter 6 and the specific support requirements identified in Chapter 7. We present our method within the ecology of existing methodologies and describe how it is a bridge to design a system to work effectively as a teammate.

The fifth major contribution of this thesis is the Interdependence Analysis (IA) Table (Chapter 8.1). This is a design and analysis tool to be used in conjunction with the Coactive Design method. It is a simple, visual way to enumerate the alternative ways by which combinations of team members can achieve a goal. If a system is to be resilient and deal with a demand for "a shift of processes, strategies and coordination" (Woods & Hollnagel, 2006, p. 22), there must be alternative processes, strategies and coordination. The IA Table enables designers to discover alternatives and understand how to support them in their systems. Based on the alternatives the designer chooses to support, the IA Table helps identify the independence relationships that must be supported for that relationship to be effective. This includes determining specific observability, predictability and directability requirements needed to support those relationships. Since design is always an iterative process, the IA Table supports this and helps understand the impact design changes might have on both individual and team performance.

Summarizing the contributions we answer our key question. Coactive Design is an approach that enables a developer to design a system to work effectively as a teammate. By following the Coactive Design method and using the IA Table, designers have a way to ground the high-level teamwork concepts into design specifications and requirements. These specifications are based on three key team capabilities: observability, predictability, and directability. Since the purpose of the IA Table is to identify the requirements necessary to support the desired interdependent relationships, it guides a designer to find alternatives that provide flexibility. This flexibility will add resilience to the final system.

## 12.2 Future Work

As is often the case in life, the more you learn, the more you learn there is more to learn. Coactive Design provides a step toward improving the resilience of human-machine systems, however, many issues relating to the problem of how to get humans and machines to work together as a resilient system are still open. Exploiting the full potential of these types of systems remains an unfinished task for the related research communities.

Beginning with the claims of this thesis, we explore some of the potential future work. We start with the claim that *resilience benefits from a teamwork infrastructure for exploiting interdependence in human-machine systems.* To make this claim more precise, we suggest to focus on the development of metrics for evaluation and to develop benchmark tasks that allow for systematic and controlled experiments.

Developing appropriate metrics is challenging. How do you measure the benefit? How much benefit is provided? Is there a methodical way to evaluate the cost-benefit relationship between the effort required to support a particular type of interdependence and its associated benefit? Do aspects of the teamwork infrastructure conflict or interfere with each other reducing anticipated benefits? Our claims are based on the rudimentary metric of completion or survival. While this is clearly a critical metric, more nuanced measures could lead to additional insights.

The complexity of teamwork makes it important to evaluate for the varying metrics in a controlled manner. This leads to several more research questions. How do you evaluate the benefits of teamwork in a controlled manner? How does the benefit vary across different ways of supporting interdependence? What are good bench mark tasks? How do the benefits scale with team size? Controlled evaluations will be an important part of collecting more detailed performance data. The joint activity testbed was a way to study teamwork in a controlled environment. We demonstrated the impact of failing to address interdependence.

An obvious follow-on is to add support for interdependence and validate that it mitigates the problems that resulted in performance degradation in our experiment. Some other options include studying the impact of team size, the difference based on type or amount of interdependence, and varying across different domains. The last research area is most challenging as it involves extending beyond the current BW4T capabilities. It would involve identifying similar domains that are simple enough to analyze, yet complex enough to be interesting. The BW4T testbed has already been used by other researchers (Harbers, Jonker, & Riemsdijk, 2012; Savarimuthu & Winikoff, 2013) and has been integrated into university level coursework [24]. Having a variety of proven joint activity testbeds would be beneficial to future research.

In our case studies, we focused on building new types of systems and our validation criterion was successful resilient performance. A controlled formal experiment would be a natural follow-on to this work. The main challenge would be what to compare it to. There are no other open source systems that one could draw upon for comparison. Even though there were twenty-five other VRC teams, the unstable nature of agile development makes it unlikely that any of these systems still are in working order. The best approach may be to enable or inhibit the features of an existing system, like IHMC's, that we claim add value.

Considering the claim that *interdependence is an effective basis for a design and analysis model of human-machine systems* we can develop some associated future research questions. Our starting point with Coactive Design involves an understanding of interdependence. Is this understanding complete? Are there other aspects to interdependence that are not captured by our description? This understanding will likely need to evolve over further analysis to reflect a more nuanced understanding of interdependence and its role in teamwork. Additionally, we presented observability, predictability, and directability as the three key team capabilities that are needed for effective human-machine collaboration. While we are convinced these three team capabilities are core components to any system, there may be other capabilities beyond these three that need to be included. Another area that will likely need to be extended is the color-coding scheme of the Interdependence Analysis Table. We used four categories that were valuable based on our experience. However, these four categories and their associated color schemes will need to be extended or modified as additional relevant categories are identified. These new categories will need to be analyzed to determine how they help describe the system, similar to the way we showed identification of brittleness, hard constraints and soft constraints. As more researchers investigate Coactive Design, it will propel the advancement of the underlying theory.

[24] See 2012-2013First-year BSc Course Multi-Agent Systems offered by M. Birna Riemsdijk and Koen Hindriks (http://mmi.tudelft.nl/trac/goal/wiki/Education accessed on 27 FEB 2014)

Future work can also come from reaching outside the scope of this thesis. It would be interesting to see if our extensions to the ideas from organizational theory and human-human team studies could propagate back into those communities.

Another interesting potential area is hardware design. In this thesis, the case studies were based on existing hardware with limited ability to modify these hardware platforms. Just as software infrastructure limitations can inhibit performance, hardware limitations can be equally challenging. An interesting follow-on area of research would be application of Coactive Design to the hardware design process. We have made an initial attempt in this area by applying Coactive Design to the hardware design of IHMC's exoskeleton. The approach helped highlight some design deficiencies and the anecdotal experience indicated that there is a potential benefit to applying Coactive Design to hardware development.

The most exciting future work is the work we cannot yet envision. As we build systems to be more coactive, we will inevitably uncover new ways to do things. The introduction of technologies that provide a new way to do something, such as cell phones and the Google search engine, did not just provide a new way to do something. They opened up a whole new world of possibilities that transformed our world. It is our hope that viewing the world through a coactive lens may someday allow robotic systems to fulfill their idealistic roles that have inhabited the imagination of humankind since their inception.

## References

Adams, J. A., Humphrey, C. M., Goodrich, M. A., Cooper, J. L., Morse, B. S., Engh, C., & Rasmussen, N. (2009). Cognitive task analysis for developing unmanned aerial vehicle wilderness search support. *Journal of Cognitive Engineering and Decision Making*, *3*(1), 1–26.

Allen, J. E., Guinn, C. I., & Horvitz, E. (1999). Mixed-Initiative Interaction. *IEEE Intelligent Systems*, *14*(5), 14–23. doi:http://dx.doi.org/10.1109/5254.796083

Annett, J. (2003). Hierarchical Task Analysis. In *Handbook of Cognitive Task Design* (pp. 17–35). London: Lawrence Erlbaum Associates.

Arkin, R. C. (1998). *Behavior-based robotics*. MIT press.

Bainbridge, L. (1983). Ironies of automation. *Automatica*, *19*(6), 775–779. doi:http://dx.doi.org/10.1016/0005-1098(83)90046-8

Baker, D. P., Day, R., & Salas, E. (2006). Teamwork as an Essential Component of High-Reliability Organizations. *Health Services Research*, *41*(4p2), 1576–1598. doi:10.1111/j.1475-6773.2006.00566.x

Beer, J. M., Fisk, A. D., & Rogers, W. A. (2012). *Toward a Psychological Framework for Levels of Robot Autonomy in Human - Robot Interaction*. (HFA-TR-1204). Atlanta, GA: Georgia Institute of Technology, School of Psychology, Human Factors and Aging Laboratory.

Blackhurst, J. L., Gresham, J. S., & Stone, M. O. (2011, October). The Autonomy Paradox. *Armed Froces Journal*, 20–40. Retrieved from http://www.armedforcesjournal.com/2011/10/7604038

Bleicher, A. (2010). The Gulf Spill's Lessons for Robotics. *IEEE Specturm Special Report*, *47*(8), 9–11.

Bradshaw, J. M., Acquisti, A., Allen, J., Breedy, M., Bunch, L., Chambers, N., Galescu, L., Goodrich, M., Jeffers, R., Johnson, M., Jung, H., Kulkarni, S., Lott, J., Olsen, D., Sierhuis, M., Suri, N., Taysom, W., Tonti, G., Uszok, A., et al. (2004). Teamwork-centered autonomy for extended human-agent interaction in space applications. *In Proceedings of the AAAI Spring Symposium*. AAAI Press.

Bradshaw, J. M., Carvalho, M., Bunch, L., Eskridge, T., Feltovich, P., Johnson, M., & Kidwell, D. (2012). Sol: An Agent-Based Framework for Cyber Situation Awareness. *KI - Künstliche Intelligenz*, *26*(2), 127–140. Retrieved from http://dx.doi.org/10.1007/s13218-012-0179-2

Bradshaw, J. M., Dignum, V., Jonker, C. M., & Sierhuis, M. (2012). Introduction to Special Issue on Human-Agent-Robot Teamwork. *IEEE Intelligent Systems*, *27*, 8–13.

Bradshaw, J. M., Feltovich, P. J., & Johnson, M. (2011). Human-Agent Interaction. In G. Boy (Ed.), *Handbook of Human-Machine Interaction* (pp. 293–302). Ashgate.

Bradshaw, J. M., Feltovich, P. J., Jung, H., Kulkarni, S., Taysom, W., & Uszok, A. (2004). Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction. In M. Klusch & G. Weiss (Eds.), *Agents and Computational Autonomy* (Vol. 2969, pp. 17–39). Berlin / Heidelberg: Springer.

Bradshaw, J. M., Jung, H., Kulkarni, S., Johnson, M., Feltovich, P., Allen, J., Bunch, L., Chambers, N., Galescu, L., Jeffers, R., Suri, N., Taysom, W., & Uszok, A. (2005). Kaa: policy-based explorations of a richer model for adjustable autonomy. *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*. The Netherlands: ACM.

Bradshaw, J. M., Jung, H., Kulkarni, S., Johnson, M., Feltovich, P., Allen, J., Bunch, L., Chambers, N., Galescu, L., Jeffers, R., Suri, N., Taysom, W., & Uszok, A. (2008). Toward Trustworthy Adjustable Autonomy and Mixed-Initiative Interaction in KAoS. Springer. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary10.1.1.108.1619

Bradshaw, J. M., Sierhuis, M., Acquisti, A., Feltovich, P., Hoffman, R., Jeffers, R., Prescott, D., Siuri, N., Uszok, A., & Van Hoof, R. (2003). Adjustable autonomy and human-agent teamwork in practice:An interim report on space applications. In H. Hexmoor, R. Falcone, & C. Castelfranchi (Eds.), *Agent Aunomy* (pp. 243–280). Boston: Kluwer Academic Publishers. Retrieved from http://www.loc.gov/catdir/toc/fy036/2003040054.html

Breazeal, C., Gray, J., Hoffman, G., & Berlin, M. (2004). Social robots : Beyond tools to partners. *Proceedings of the Thirteenth IEEE Workshop on Robot and Human Interactive Communication (Ro-Man04)*. Kurashiki, Japan.

Brennan, S. E. (1998). The grounding problem in conversations with and through computers. *Social and Cognitive Approaches to Interpersonal Communication*, 201–225.

Brisset, P., & Hattenberger, G. (2008). Multi-UAV Control with the Paparazzi System. In *The first conference on Humans Operating Unmanned Systems (HUMOUS'08)*. Brest, France.

Brookshire, J., Singh, S., & Simmons, R. (2004). Preliminary Results in Sliding Autonomy for Coordinated Teams. In *Proceedings of The 2004 Spring Symposium Series*.

Burke, J. L., Murphy, R. R., Coovert, M. D., & Riddle, D. L. (2004). Moonlight in Miami: a field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Hum.-Comput. Interact.*, *19*(1), 85–116.

Cantrell, R., Scheutz, M., Schermerhorn, P., & Wu, X. (2010). Robust spoken instruction understanding for HRI. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on* (pp. 275–282).

Carff, J., Johnson, M., El-Sheikh, E. M., & Pratt, J. E. (2009). Human-robot team navigation in visually complex environments. *International Conference on Intelligent Robots and Systems (IROS 2009)*. St. Louis, MO.

Castelfranchi, C. (2000). Founding Agents' "Autonomy" on Dependence Theory. In H. W. (Ed.), *The 14th European Conference on Artificial Intelligence (ECAI )* (pp. 353–357). Berlin, Germany: IOS Press.

Cervin, B., Mills, C., & Wünsche, B. C. (2004). A 3D interface for an unmanned aerial vehicle. *SE Stage*, *4*.

Chen, J. Y. C., Haas, E. C., & Barnes, M. J. (2007). Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and Reviews, Vol.37, No. 6*, *37*(6), 1231–1245.

Chipalkatty, R., Droge, G., & Egerstedt, M. B. (2013). Less Is More: Mixed-Initiative Model-Predictive Control With Human Inputs. *Robotics, IEEE Transactions On*, *29*(3), 695–703.

Christoffersen, K., & Woods, D. D. (2002). How to Make Automated Systems Team Players. *Advances in Human Performance and Cognitive Engineering Research*, *2*, 1–12.

Clark, H. H. (1996). *Using language* (p. xi, 432 p.). Cambridge [England] ; New York: Cambridge University Press. Retrieved from http://www.loc.gov/catdir/toc/cam023/95038401.html

Clark, H. H., & Brennan, S. (1991). Grounding in communication. *Perspectives on Socially Shared Cognition*, 127–149.

Cohen, P. R., & Levesque, H. J. (1991). Teamwork. *Nous*, *25*(4), 487–512.

Cooper, J. L. (2007). *Supporting Flight Control for UAV-assisted Wilderness Search and Rescue Through Human Centered Interface Design*. Brigham Young University. Department of Computer Science. Retrieved from http://books.google.com/books?id=fDtgQwAACAAJ

Covey, S. R. (1989). *The 7 Habits of Highly Effective People*. New York: Free Press.

Crandall, B., & Klein, G. (2006). *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. Bradford Books.

Crandall, J. W., & Cummings, M. L. (2007). Developing performance metrics for the supervisory control of multiple robots. *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*. Arlington, Virginia, USA: ACM. doi:http://doi.acm.org/10.1145/1228716.1228722

Crandall, J. W., & Goodrich, M. A. (2002). Principles of adjustable interactions. *AAAI Fall Symposium Human-Robot Interaction Workshop*. North Falmouth, MA.

Crandall, J. W., Goodrich, M. A., Olsen, D. R. J., & Nielsen, C. W. (2005). Validating human-robot interaction schemes in multitasking environments.

*Systems, Man and Cybernetics*, *35*(4), 438–449. doi:10.1109/TSMCA.2005.850587

Cuevas, H. M., Fiore, S. M., Caldwell, B. S., & Strater, L. (2007). Augmenting team cognition in human-automation teams performing in complex operational environments. *Aviation, Space, and Environmental Medicine*, *78*(Supplement 1), B63–B70.

Cummings, M. L., da Silva, F. B., & Scott, S. D. (2007). *Design Methodology for Unmanned Aerial Vehicle (UAV) Team Coordination*. MIT Humans and Automation Laboratory.

Defense Science Board. (2012). *Defense Science Board Task Force Report: The Role of Autonomy in DoD Systems*. Retrieved from http://www.dtic.mil/get-tr-doc/pdf?AD=ADA566864

Dempsey, M. (2010). Us army unmanned aircraft systems roadmap 2010-2035. *Federation Of American Scientists*.

Department of Defense. (2007). *Unmanned Systems Roadmap: 2007-2032*. Retrieved from https://www.google.com/#q=unmanned+systems+roadmap+2007

Department of Defense. (2013). *Unmanned Systems Integrated Roadmap, FY 2013–2038*. Retrieved from www.defense.gov/pubs/DOD-USRM-2013.pdf?

Dias, M. B., Kannan, B., Browning, B., Jones, E., Argall, B., Dias, M. F., Zinck, M. B., Veloso, M. M., & Stentz, A. (Tony). (2008). Sliding Autonomy for Peer-To-Peer Human-Robot Teams. tech. report CMU-RI-TR-08-16, Robotics Institute, Carnegie Mellon University: Robotics Institute.

Diggelen, J. van, Bradshaw, J., Johnson, M., Uszok, A., & Feltovich, P. (2009). Implementing Collective Obligations in Human-Agent Teams using KAoS Policies. *Proceedings of Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*. Springer.

Dorais, G., & Kortenkamp, D. (2001). Designing Human-Centered Autonomous Agents. *Revised Papers from the PRICAI 2000 Workshop Reader, Four Workshops Held at PRICAI 2000 on Advances in Artificial Intelligence*. Springer-Verlag.

Dragan, A. D., Lee, K. C. T., & Srinivasa, S. S. (2013). Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction* (pp. 301–308). Piscataway, NJ, USA: IEEE Press. Retrieved from http://dl.acm.org/citation.cfm?id=2447556.2447672

Drury, J. L., Richer, J., & Rackliffe, N. (2006). Comparing situation awareness for two unmanned aerial vehicle human interface approaches. In *In Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*.

Endsley, M R, Bolté, B., & Jones, D. G. (2003). *Designing for Situation Awareness: An approach to user-centered design*. Taylor & Francis.

Endsley, Mica R. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, *42*(3), 462–492.

Enes, A., & Book, W. (2010). Blended shared control of Zermelo's navigation problem. In *American Control Conference (ACC), 2010* (pp. 4307–4312).

Feltovich, P. J., Bradshaw, J. M., Clancey, W. J., & Johnson, M. (2007). Toward an Ontology of Regulation: Socially-Based Support for Coordination in Human and Machine Joint Activity. In G. O'Hare, M. O'Grady, A. Ricci, & O. Dikenelli (Eds.), *Engineering Societies in the Agents World VII* (Vol. Lecture No, pp. 175–192). Heidelberg, Germany: Springer.

Feltovich, P. J., Bradshaw, J. M., Clancey, W. J., Johnson, M., & Bunch, L. (2008). Progress Appraisal as a Challenging Element of Coordination in Human and Machine Joint Activity. In A. Artikis, G. M. P. O'Hare, K. Stathis, & G. Vouros (Eds.), *Engineering Societies in the Agents World VIII* (Vol. Lecture No, pp. 124–141). Heidelberg, Germany: Springer.

Fiore, S. M. (2008). Interdisciplinarity as teamwork how the science of teams can inform team science. *Small Group Research*, *39*(3), 251–277.

Fitts, P. M. (1951). *Human engineering for an effective air-navigation and traffic-control system* (p. xii, 84 p.). Washington,: National Research Council, Division of Anthropology and Psychology, Committee on Aviation Psychology.

Fitts, P. M., & Posner, M. I. (1967). *Human performance. Basic concepts in psychology series* (p. x, 162 p.). Belmont, Calif.,: Brooks/Cole Pub. Co.

Fong, T. W. (2001). Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.

Giesler, B., Salb, T., Steinhaus, P., & Dillmann, R. (2004). Using Augmented Reality to Interact with an Autonomous Mobile Platform. In *IEEE Conference on Robotics and Automation (ICRA)*. New Orleans.

Goodrich, M. A. (2004). *Using models of cognition in HRI evaluation and design*. Defense Technical Information Center.

Goodrich, M. A., & Schultz, A. C. (2007). Human-robot interaction: a survey. *Foundations and Trends in Human-Computer Interaction*, *1*(3), 203–275. doi:http://dx.doi.org/10.1561/1100000005

Guzzo, R. A., & Salas, E. (1995). *Team effectiveness and decision making in organizations*. *The Jossey-Bass management series* (1st ed., p. xxiv, 414 p.). San Francisco: Jossey-Bass.

Hancock, P. A., Jagacinski, R. J., Parasuraman, R., Wickens, C. D., Wilson, G. F., & Kaber, D. B. (2013). Human-Automation Interaction Research Past, Present, and Future. *Ergonomics in Design: The Quarterly of Human Factors Applications*, *21*(2), 9–14.

Harbers, M., Jonker, C. M., & Riemsdijk, M. B. Van. (2012). Enhancing Team Performance through Effective Communication. In *HART workshop*. Boston, MA.

Hoffman, R. R, Ford, K. M., & W, C. J. (2000). *The Handbook of Human-Centered Computing*. Institute of Human Machine Cognition, University of West Florida, Pensacola, FL.

Hoffman, Robert R, & Deal, S. V. (2008). Influencing versus Informing Design, Part 1: A Gap Analysis. *IEEE Intelligent Systems*, *23*(5), 78–81.

Hoffman, Robert R, Hayes, P., Ford, K. M., & Bradshaw, J. M. (Eds.). (2012). *Collected Essays on Human-Centered Computing, 2001-2011*. New York City: IEEE Press.

Hoffman, Robert R, & Woods, D. D. (2011). Beyond Simon's Slice: Five Fundamental Trade-Offs that Bound the Performance of Macrocognitive Work Systems. *Intelligent Systems, IEEE*, *26*(6), 67–71. doi:10.1109/MIS.2011.97

Humphrey, C. M., Motter, S. R., Adams, J. A., & Gonyea, M. (2009). A human eye like perspective for remote vision. *Proceeding of the 2009 IEEE International Conference on System, Man and Cybernetics San Antonio, TX, USA - October 2009*, (October), 1712–1717.

Jennings, N. R. (1996). Coordination Techniques for Distributed Artificial Intelligence. In G. M. P. O'Hare & N. R. Jennings (Eds.), *Foundations of Distributed Artificial Intelligence* (pp. 187–210). New York, NY: John Wiley & Sons, Inc.

Johnson, M., Bradshaw, J., Feltovich, P., Jonker, C., van Riemsdijk, B., & Sierhuis, M. (2011). The Fundamental Principle of Coactive Design: Interdependence Must Shape Autonomy. In M. De Vos, N. Fornara, J. Pitt, & G. Vouros (Eds.), *Coordination, Organizations, Institutions, and Norms in Agent Systems VI* (Vol. 6541, pp. 172–191). Springer Berlin / Heidelberg. doi:10.1007/978-3-642-21268-0_10

Johnson, M., Bradshaw, J. M., Feltovich, P. J., Hoffman, R. R., Jonker, C. M., van Riemsdijk, M. B., & Sierhuis, M. (2011). Beyond Cooperative Robotics: The Central Role of Interdependence in Coactive Design. *IEEE Intelligent Systems*, *26*, 81–88. Retrieved from http://doi.ieeecomputersociety.org/10.1109/MIS.2011.47

Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., van Riemsdijk, M. B., & Sierhuis, M. (2014). Coactive design: Designing support for interdependence in joint activity. *Human Robot-Interaction*.

Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C., van Riemsdijk, B., & Sierhuis, M. (2012). Autonomy and interdependence in human-agent-robot teams. *Intelligent Systems, IEEE*, *27*(2), 43–51. doi:10.1109/MIS.2012.1

Johnson, M., Carff, J., & Pratt, J. (2012). Coactive Design For Human-MAV Team Navigation. In *International Micro Aerial Vehicle Conference and Competition*. Germany.

Johnson, M., Jonker, C., Riemsdijk, B. van, Feltovich, P. J., & Bradshaw, J. M. (2009). Joint Activity Testbed: Blocks World for Teams (BW4T). *Engineering Societies in the Agents World X*. Springer.

Jonker, C. M., Van Riemsdijk, M. B., & Vermeulen, B. (2011). Shared mental models. In *Coordination, Organizations, Institutions, and Norms in Agent Systems VI* (pp. 132–151). Springer.

Kaber, D. B., & Riley, J. M. (1999). Adaptive Automation of a Dynamic Control Task Based on Secondary Task Workload Measurement. *International Journal of Cognitive Ergonomics*, *3*(3), 169–187. doi:10.1207/s15327566ijce0303_1

Kaber, D. B., Riley, J. M., Tan, K.-W., & Endsley, M. R. (2001). On the design of adaptive automation for complex systems. *International Journal of Cognitive Ergonomics*, *5*(1), 37–57.

Kaminka, G. A., Elmaliach, Y., Frenkel, I., Glick, R., Kalech, M., & Shpigelman, T. (2004). Towards a Comprehensive Framework for Teamwork in Behavior-Based Robots. IOS Press.

Kidd, P. T. (1992). Design of human-centered robotic systems. In M. Rahimi & W. Karwowski (Eds.), *Human-Robot Interaction* (pp. 225–241). Taylor & Francis.

Klein, G., Feltovich, P. J., Bradshaw, J. M., & Woods, D. D. (2005). Common Ground and Coordination in Joint Activity. In K. R. B. William B. Rouse (Ed.), *Organizational Simulation* (pp. 139–184). Retrieved from http://dx.doi.org/10.1002/0471739448.ch6

Klein, G., Woods, D. D., Bradshaw, J. M., Hoffman, R. R., & Feltovich, P. J. (2004). Ten Challenges for Making Automation a "Team Player" in Joint Human-Agent Activity. *IEEE Intelligent Systems*, *19*(6), 91–95. doi:http://dx.doi.org/10.1109/MIS.2004.74

Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (Vol. 3, pp. 2149–2154 vol.3). doi:10.1109/IROS.2004.1389727

Kortenkamp, D. (2001). Designing an Architecture for Adjustably Autonomous Robot Teams. *Revised Papers from the PRICAI 2000 Workshop Reader, Four Workshops Held at PRICAI 2000 on Advances in Artificial Intelligence*. Springer-Verlag.

Larson, C. E., & LaFasto, F. M. J. (1989). *Teamwork : what must go right, what can go wrong*. *Sage series in interpersonal communication* (p. 150 p.). Newbury Park, Calif.: SAGE Publications.

Lesser, V. R. (1991). A retrospective view of FA/C distributed problem solving. *Systems, Man and Cybernetics, IEEE Transactions On*, *21*(6), 1347–1362.

Lorite, S., Muñoz, A., Tornero, J., Ponsa, P., & Pastor, E. (2013). Supervisory Control Interface Design for Unmanned Aerial Vehicles through GEDIS-UAV. In M. Kurosu (Ed.), *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments SE - 26* (Vol. 8004, pp. 231–240). Springer Berlin Heidelberg. doi:10.1007/978-3-642-39232-0_26

Macbeth, J. C., Cummings, M. L., Bertuccelli, L. F., & Surana, A. (2012). Interface Design for Unmanned Vehicle Supervision through Hybrid Cognitive Task Analysis. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 56, pp. 2344–2348).

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Comput. Surv.*, *26*(1), 87–119. doi:http://doi.acm.org/10.1145/174666.174668

March, J. G., Simon, H. A., & Guetzkow, H. S. (1993). *Organizations* (2nd ed., p. ix, 287 p.). Cambridge, Mass., USA: Blackwell.

McCarley, J. S., & Wickens, C. D. (2005). *Human factors implications of UAVs in the national airspace*. University of Illinois at Urbana-Champaign, Aviation Human Factors Division.

McIntyre, R. M., & Salas, E. (1995). Measuring and managing for team performance: Emerging principles from complex environments. In *Team Effectiveness and Decision Making in Organizations* (pp. 149–203). San Francisco, CA: Jossey-Bass.

Michaud, F., Boissy, P., Labonte, D., Briere, S., Perreault, K., Corriveau, H., Grant, A., Lauria, M., Cloutier, R., Roux, M.-A., Iannuzzi, D., Royer, M.-P., Ferland, F., Pomerleau, F., & Letourneau, D. (2010). Exploratory design and evaluation of a homecare teleassistive mobile robotic system. *Mechatronics*, *20*(7), 751–766. doi:http://dx.doi.org/10.1016/j.mechatronics.2010.01.010

Miller, C. (2012). Frameworks for Supervisory Control: Characterizing Relationships with Uninhabited Vehicles. *Journal of Human-Robot Interaction*, *1*(2), 183–201.

Murphy, R., Casper, J., Micire, M., & Hyams, J. (2000). Mixed-initiative Control of Multiple Heterogeneous Robots for USAR. Technical report, University of South Florida.

Murphy, R. R., & Burke, J. L. (2008). From remote tool to shared roles. *Robotics Automation Magazine, IEEE*, *15*(4), 39–49. doi:10.1109/MRA.2008.929928

Myers, K. L., & Morley, D. N. (2001). Directing Agent Communities: An Initial Framework. In *Proceedings of the IJCAI Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*. Seattle, WA. Retrieved from http://www.ai.sri.com/pubs/files/840.pdf

Nielsen, C. W., Goodrich, M. A., & Ricks, R. W. (2007). Ecological Interfaces for Improving Mobile Robot Teleoperation. *Robotics, IEEE Transactions On*, *23*(5), 927–941. doi:10.1109/TRO.2007.907479

Norman, D. A. (1988). *Design of everyday things*. Basic Books.

Norman, D. A. (1990). The “problem” of automation: Inappropriate feedback and interaction, not “over-automation.” In D. E. Broadbent, A. Baddeley, & J. T. Reason (Eds.), *Human factors in hazardous situations* (pp. 585–593). Oxford University Press.

Norman, D. A. (1991). Cognitive artifacts. In J. M. Carroll (Ed.), *Designing interaction: Psychology at the human-computer interface* (pp. 17–38). New York: Cambridge University Press.

Olsen, D. R., & Goodrich, M. (2003). "Metrics for Evaluating Human-Robot Interaction. In *Proceedings of PERMIS*.

Parasuraman, R., Sheridan, T., & Wickens, C. (2000). A model for types and levels of human interaction with automation. *Systems, Man and Cybernetics, Part A, IEEE Transactions On*, *30*(3), 286–297. Retrieved from http://dx.doi.org/10.1109/3468.844354

Ponsa, P., & Díaz, M. (2007). Creation of an Ergonomic Guideline for Supervisory Control Interface Design. In *Proceedings of the 7th International Conference on Engineering Psychology and Cognitive Ergonomics* (pp. 137–146). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1784197.1784213

Quigley, M., Goodrich, M. A., & Beard, R. W. (2004). Semi-autonomous human-UAV interfaces for fixed-wing mini-UAVs. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (Vol. 3, pp. 2457–2462).

Reitmayr, G., & Schmalstieg, D. (2004). Collaborative augmented reality for outdoor navigation and information browsing. In *Proceedings of the Second Symposium on Location Based Services and TeleCartography* (pp. 53–62). TU Wien.

Saavedra, R., Earley, P. C., & Van Dyne, L. (1993). Complex interdependence in task-performing groups. *Journal of Applied Psychology*, *78*(1), 61.

Salas, E., Bowers, C. A., & Edens, E. (2001). *Improving teamwork in organizations : applications of resource management training* (p. ix, 356 p.). Mahwah, N.J.: Lawrence Erlbaum Associates.

Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On Teams, Teamwork, and Team Performance: Discoveries and Developments. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(3), 540–547. doi:10.1518/001872008X288457

Salas, E., Dickinson, T. L., Converse, S. A., & Tannenbaum, S. I. (1992). Toward an understanding of team performance and training. Teams: Their training and performance. *In Swezey R. W.& Salas E. (Eds.), Teams: Their Training and Performance, NJ: Ablex, pp.3-29.Sandstorm E, De Meuse KP, Futrell D.,(1990)"Work Terms: Applications and Effectiveness" American Psychologist.;45(2):120–33.* Retrieved from citeulike-article-id:7187620

Salas, E., & Fiore, S. M. (2004). Team cognition understanding the factors that drive process and performance. Washington, DC: American Psychological Association. Retrieved from http://www.loc.gov/catdir/toc/ecip0411/2003026064.html

Salas, E., Rosen, M. A., Burke, C. S., Goodwin, G. F., Fiore, S. and E. P., & Press., (pp. 439-453). New York: Cambridge University. (2006). The making of a dream team: when expert teams do best. In K. A. Ericsson, N. Charness, P. J. Feltovich, & R. R. Hoffman (Eds.), *The Cambridge Handbook of Expertise*.

Sarter, N. B., & Woods, D. D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 5–19.

Savarimuthu, S., & Winikoff, M. (2013). Mutation Operators for the Goal Agent Language. In M. Cossentino, A. Fallah Seghrouchni, & M. Winikoff (Eds.), *Engineering Multi-Agent Systems SE - 14* (Vol. 8245, pp. 255–273). Springer Berlin Heidelberg. doi:10.1007/978-3-642-45343-4_14

Schraagen, J. M., Chipman, S. F., & Shalin, V. L. (2009). *Cognitive Task Analysis*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Schurr, N., Okamoto, S., Maheswaran, R. T., Scerri, P., & Tambe, M. (2005). Evolution of a Teamwork Model. In *COGNITION AND MULTI-AGENT INTERACTION: FROM COGNITIVE MODELING TO SOCIAL SIMULATION* (pp. 307–327). Cambridge University Press.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control* (p. xx, 393 p.). Cambridge, Mass.: MIT Press.

Sheridan, T. B. (2011). Adaptive automation, level of automation, allocation authority, supervisory control, and adaptive control: Distinctions and modes of adaptation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions On*, *41*(4), 662–667.

Sheridan, T. B., & Verplank, W. (1978). Human and Computer Control of Undersea Teleoperators. Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT.

Sierhuis, M. (2007). "It's not just goals all the way down" - "It's activities all the way down". *Engineering Societies in the Agents World VII, 7th International, Workshop, ESAW 2006*. Dublin, Ireland.

Sierhuis, M., Bradshaw, J. M., Acquisti, R., Hoof, R. Van, & Jeffers, R. (2003). Human-Agent Teamwork and Adjustable Autonomy in Practice. *Proceedings of the Seventh International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS*.

Smith-Jentsch, K. A., Baker, D. P., Salas, E., & Cannon-Bowers, J. A. (2001). Uncovering Differences in Team Competency Requirements. In E Salas, C. A. Bowers, & E. Edens (Eds.), *Improving Teamwork in Organizations* (pp. 31–54). Lawrence Erlbaum Associates.

Smith-Jentsch, K. A., Zeisig, R. L., Acton, B., & McPherson, J. A. (1998). Team dimensional training: A strategy for guided team self-correction. In J. A. Cannon-Bowers & E. Salas (Eds.), *Making decisions under stress: Implications for individual and team training* (pp. 271–297). Washington, DC: American Psychological Association.

Stubbs, K., Hinds, P., & Wettergreen, D. (2007). Autonomy and common ground in human-robot interaction: A field study. *IEEE Intelligent Systems*, (Special Issue on Interacting with Autonomy), 42–50.

Sycara, K. (2002). Integrating agents into human teams. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 46, pp. 413–417).

Sycara, K., & Sukthankar, G. (2006). *Literature Review of Teamwork Models*. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.

Tambe, M. (1997). Towards Flexible Teamwork. *JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH*, *7*, 83–124.

Thibaut, J. W., & Kelley, H. H. (1959). *The Social Psychology of Groups* (p. xii, 313 p.). New York,: Wiley.

Thompson, J. D. (1967). *Organizations in Action; Social Science Bases of Administrative Theory* (p. xi, 192 p.). New York,: McGraw-Hill.

United States Air Force. (2010). *Technology Horizons. A Vision for Air Force Science & Technology During 2010-2030.*

Woods, D. D., & Branlat, M. (2010). Basic Patterns in How Adaptive Systems Fail. In J. Pariès, E. Hollnagel, J. Wreathall, & D. D. Woods (Eds.), *Resilience Engineering in Practice: A Guidebook* (pp. 127–144). Ashgate.

Woods, D. D., & Hollnagel, E. (2006). *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering*. Taylor & Francis.

Woods, D. D., & Sarter, N. B. (1997). Automation Surprises. In G. Salvendy (Ed.), *Handbook of Human Factors & Ergonomics* (second.). Wiley.

# Samenvatting

Deze thesis komt voort uit de frustratie die wij hebben ervaren bij het ontwerpen van onze eigen autonome systemen en vanuit autonome systemen die we gezien hebben van anderen. Bestaande autonome systemen kunnen niet goed omgaan met de onzekerheden, ambiguiteïten, en onverwachte situaties die zich in de werkelijkheid nu eenmaal voordoen. In de Unmanned Systems Integrated Roadmap van het ministerie van Defensie van de USA (2013, p.29) staat bijvoorbeeld dat bijna alle onbemande systemen om directe besturing vragen voor zowel basale handelingen als voor gedrag te maken heeft met communicatie, inzet van personeel, en de effectiviteit van het systeem. Dit wordt in een rapport van de Defense Science Board (DSB) toegeschreven aan het probleem dat autonome systemen niet goed om kunnen gaan met nieuwe situaties (2012, p. 58).

Ons doel is autonome systemen effectiever te maken. Dit betekent dat ze een groot herstellingsvermogen moeten hebben. Herstellingsvermogen gaat niet over optimaal gedrag, het gaat over overleven en het kunnen uitvoeren van hun taak. David Woods beschrijft het als het vermogen van systemen om een onverwachte verstoring te kunnen herkennen en om hun gedrag hieraan aan te passen. Dit vraagt om een nieuw model van wat een goed functionerend systeem is, en om een verandering van bestaande processen, strategieën, en coordinatie (2006, p. 22). Zijn beschrijving gaat in op twee fundamentele aspecten van herstellingsvermogen: het herkennen van problemen en het bieden van flexibele alternatieven om met die problemen om te gaan. Daarmee wordt de onderzoeksvraag dus: hoe ontwerp je een systeem dat een goed herstellingsvermogen heeft?

Coactive Design breekt met traditionele ontwerpmethodes door zich te richten op het effectief omgaan met de onderlinge afhankelijkheden tussen teamleden van een mens-machine systeem (Johnson, Bradshaw, Feltovich, Jonker, et al., 2011). In de resulterende mens-machine systemen kunnen teamleden problemen herkennen en als team hun gedrag hieraan aanpassen. Als een team om kan gaan met een scala aan onderlinge afhankelijkheidsrelaties dan maakt dat het team flexibel. Flexibiliteit geeft het systeem herstellingsvermogen doordat alternatieve manieren beschikbaar zijn om onverwachte situaties te herkennen en het hoofd te bieden.

Coactive Design zoals we beschreven in dit proefschrift, biedt vijf belangrijke bijdragen: 1) een nieuw ontwerpperspectief gebaseerd op onderlinge afhankelijkheden, 2) een rijker begrip van onderlinge afhankelijkheden, 3) een nieuw model voor mens-machine systemen, 4) een nieuwe ontwerpmethode, en 5) een nieuw gereedschap, de Interdependence Analysis Table, dat de ontwerper helpt bij het systeemontwerp en -analyse. Hoofdstukken 5-8 bevatten de informatie waarmee andere ontwerpers deze methodiek toe kunnen passen op het ontwerp van hun eigen mens-machine systemen.

We hopen dat ontwerpers van mens-machine systemen het Coactive Design perspectief ervaren als verfrissend, en dat dit proefschrift een nieuw licht werpt op hun ontwerpuitdagingen. Evenzo hopen we dat ze onze methode en gereedschappen waardevol vinden in hun ontwerpproces. Met Coactive Design kunnen abstracte samenwerkingsconcepten vertaald worden in herbruikbare algoritmen en interface-elementen waarmee robots hun beoogde rol als teamlid kunnen vervullen. Onderlinge afhankelijkheden zijn belangrijk omdat deze de basis vormen voor het begrijpen van complexe systemen. Door het gebruik van de Coactive Design methode kan een ontwerper mens-machine systemen ontwikkelen welke om kan gaan met onderlinge afhankelijkheden en welke daarmee niet alleen meer basisvaardigheden, maar ook flexibiliteit en herstellingsvermogen hebben.

## About the author

Matthew Johnson was born in New York in 1970. On graduating from high school he received a full Navy scholarship to attend the University of Notre Dame. He received his Bachelor of Science in Aerospace Engineering in 1992 and proceeded to the Navy's flight school in Pensacola Florida. In 1994 he was designated a Naval Aviator and received his wings of gold. He spent ten years on active duty, completing two deployments. He flew SH-60B helicopters and was a flight instructor in the T-34C. During his last two years of active duty service he attended Texas A&M – Corpus Christi and received a Master of Science in Computer Science in 2001. After leaving active duty, he continued as a flight instructor in the Navy Reserves retiring after twenty years of service in 2012.

In 2002 Matt shifted his focus from flying to research. He began working at the Florida Institute for Human and Machine Cognition. He has worked on numerous projects including the Oz flight display for reducing the cognitive workload in the cockpit, Augmented Cognition for improving human performance, and several human-robot coordination projects for both NASA and the Department of Defense. He has worked on advanced robotic control projects such as the DARPA Little Dog project developing walking algorithms for a quadruped robot on rough terrain and the IHMC lower body humanoid developing low-gravity walking gaits for NASA. He has developed advanced information sharing systems designed to support the forward deployed soldier. Since 2009 Matt has been working on his PhD in Computer Science with TU Delft focusing on improving performance in human-machine systems. This work was done in collaboration with his colleagues at TU Delft and those at IHMC in Pensacola, Florida.