Coactive Design:

Designing Support for Interdependence in Joint Activity

Matthew Johnson, Jeffrey M. Bradshaw, Paul J. Feltovich
Florida Institute for Human and Machine Cognition (IHMC)
Pensacola, Florida, USA
and
Catholijn M. Jonker, M. Birna van Riemsdijk
Delft University of Technology, The Netherlands
and
Maarten Sierhuis
Ejenta and Nissan Research Center Silicon Valley, CA, USA

Coactive Design is a new approach to address the increasingly sophisticated roles that people and robots play as the use of robots expands into new, complex domains. The approach is motivated by the desire for robots to perform less like teleoperated tools or independent automatons and more like interdependent teammates. In this article, we describe what it means to be interdependent, why this is important, and the design implications that follow from this perspective. We argue for a human-robot system model that supports interdependence through careful attention to requirements for observability, predictability, and directability. We present a Coactive Design method and show how it can be a useful approach for developers trying to understand how to translate high-level teamwork concepts into reusable control algorithms, interface elements, and behaviors that enable robots to fulfill their envisioned role as teammates. As an example of the coactive design approach, we present our results from the DARPA Virtual Robotics Challenge, a competition designed to spur development of advanced robots that can assist humans in recovering from natural and man-made disasters. Twenty-six teams from eight countries competed in three different tasks providing an excellent evaluation of the relative effectiveness of different approaches to human-machine system design.

Keywords: coactive design, joint activity, teamwork, collaboration, interdependence, human-robot interaction, human-agent-robot teamwork

Motivation

The view of robots as teammates has grown as the field of robotics has matured. This is evidenced by the 2011 National Robotics Initiative[1], which had a goal of accelerating the development and use of robots that work beside, or cooperatively with, people. Considering robots as teammates is also prevalent in most future-looking roadmaps. This includes civilian ones such as *NextGen UAS Research Air Transportation Roadmap* from 2012 and military ones such as the *Unmanned Systems Integrated Roadmap FY2011-2036* that both refer to Manned-Unmanned (MUM) Teaming, as well as academia-generated ones such as *A Roadmap for US Robotics* from 2009,

---

[1] http://www.nsf.gov/pubs/2011/nsf11553/nsf11553.htm accessed 2013-07-12 1900CST

which refers to human-robot teams. The need to work closely with humans is also echoed by the development of general purpose service robots, e.g., the PR2 from Willow Garage, that promise to someday fulfill our long-time desires for these automated assistants. Some projected uses of this type of robot include household help, assisting the elderly or disabled, and assistance as a co-worker in various service fields (Wyrobek, Berger, Van der Loos, & Salisbury, 2008). Given this landscape for robotics uses, it should be clear that today's goals for robotics envision 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).

The goal of designing a teammate captures the expectations of future robotic systems in virtually every kind of application and the more sophisticated roles they are expected to play. But what does it mean for a system to be a teammate? Effective teamwork intuitively 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. As technology has become an increasingly prevalent part of human life, the social science communities and engineering communities have been forced together to address a system comprising both man and machine. Models of teamwork have been developed but tend to be lists of characteristics such as having clearly defined roles and effective communication (Larson & LaFasto, 1989), or of high-level behaviors such as providing periodic updates and monitoring for errors (Smith-Jentsch, Zeisig, Acton, & McPherson, 1998; 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 (Hoffman & Deal, 2008). The challenge is in translating high-level concepts such as teamwork and collaboration into implementations of such concepts within control algorithms, interface elements, and behaviors.

Background and Related Work

Much of the early work of researchers in robotics and software agents was motivated by situations in which autonomous systems were envisioned to "replace" human participation, thus minimizing the need to consider the "social" aspects of working together. Traditional approaches to human-robot systems design usually employ task decomposition and allocation. The most well-known example of this approach is supervisory control (Sheridan, 1992), in which people allocate tasks to one or more machines and then monitor their performance. 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.

To address requirements for variable task allocation in different situations, there has been interest in dynamic and adaptive function allocation—also known in slightly different forms as adjustable autonomy, dynamic task allocation, sliding autonomy, flexible autonomy, or adaptive automation. In each case, the system must decide at runtime which functions to automate and to what level of autonomy (Parasuraman, Sheridan, & Wickens, 2000). The insight these approaches provide is a need for flexibility and adaptability in a system. Our extensive work in this area (Bradshaw et al., 2003, 2004, 2005, 2008; Sierhuis et al., 2003) has given us a deep understanding of the challenges associated with these approaches (Johnson 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 (1993) point out that "one peculiar characteristic of the assignment problem…is that, if taken literally, problems of coordination are eliminated" (p. 44). This is because approaches based on allocation unrealistically tend to ignore what March et al. (1993) describe as "the contingent character of activities." 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.

Similar to many researchers in the Human-Robot Interaction (HRI) field, we have also been influenced by the social sciences. Past research on interdependence in the social sciences includes the organizational theory work of Thompson (1967) who identified the 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 research communities have developed Hierarchical Task Analysis (HTA) (Annett, 2003) as a method for identifying and decomposing complex tasks. Cognitive Task Analysis (CTA) (Crandall & Klein, 2006; Schraagen, Chipman, & Shalin, 2009) has extended this methodology to include a representation of the knowledge and reasoning required to perform tasks. Goal-Directed Task Analysis (GDTA) (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 we will discuss further on.

The most influential contributions we drew upon in developing Coactive Design came from a variety of sources. One is joint activity theory (Bradshaw, Feltovich, & Johnson, 2011; Klein, Feltovich, Bradshaw, & Woods, 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). Malone and Crowston's (1994) work, which defined coordination as managing dependencies, provided guidance, as did the principles of human-centered computing (HCC) and the challenges identified by that community (Hoffman, Hayes, Ford, & Bradshaw, 2012). Also, Collaborative Control (Fong, 2001) has been influential in our thinking. It introduced to robotics the idea that both parties may participate *simultaneously* in the same action, which correlates well with Clark's (1996) notion of participatory action.

Coactive Design

Coactive Design is a new approach to address the increasingly sophisticated roles that people and robots play as the use of robots expands into new, complex domains. The term "coactive design" was coined as a way of characterizing an approach to the design of HRI that takes interdependence as the central organizing principle among people and robots working together in joint activity (Johnson et al., 2011). Besides implying that two or more parties are participating in an activity, the term "coactive" is meant to convey the reciprocal and mutually constraining nature of actions and effects that are conditioned by coordination. Coactive Design is focused on systems where the human and machine are engaged in joint activity. The term "joint activity" describes the nature of the activity, and the term "coactive design" is about designing in a way to accomplish effective joint activity. The goal of Coactive Design is to help designers identify interdependence relationships in a joint activity so they can design systems that support these relationships, thus enabling designers to achieve the objectives of coordination, collaboration, and teamwork.

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[2]. 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[3]. 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. However, the description 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 has 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 identifying dependencies between agents for facilitating coordination. The reason interdependence is so important in all of these domains is because it is

---

[2] 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.
[3] http://dictionary.reference.com/browse/interdependent?s=t (accessed 17 November 2013).

the basis for understanding complex systems. Again, however, the concept often 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."

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.*

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.

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 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. 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.

Interdependence Relationships Must Be Complementary

In joint activity, the interdependence relationship must be complementary. As a simple example, imagine a train engine pulling a cargo car (as depicted in *Figure 1*). 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.

Figure 1. A train analogy describing the interdependence of the car and engine. The relationship is complementary, because the coupling joining the two must fit correctly and both must support the load.

Michael Goodric
Deleted: Figure

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 2*. 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 2*.

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"). 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.

Many aspects of teamwork are best described as soft interdependencies. Our observations to date suggest that good teams can often be distinguished from great ones by how well they manage soft interdependencies.
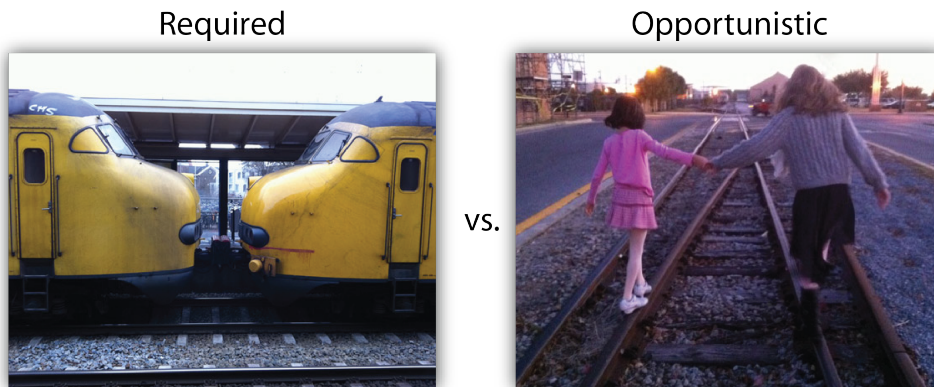

Required   Opportunistic

vs.

Figure 2. Hard (required) versus soft (opportunistic) interdependence relationships.

Interdependence Recasts Context and Defines Common Ground

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. This recasting of context creates a new regulatory constraint that is referred to as common ground (Clark & Brennan, 1991; Klein et al., 2005). Clark states that "all collective actions are built on common ground (Clark & Brennan, 1991)." It includes pertinent knowledge, beliefs and assumptions (Klein et al., 2004), but what determines pertinence? We propose that the interdependent relationships between parties determine what is relevant and thus define what common ground for a particular team should entail.

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 define what is pertinent for 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.

How to Determine Interdependence Requirements

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. 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.

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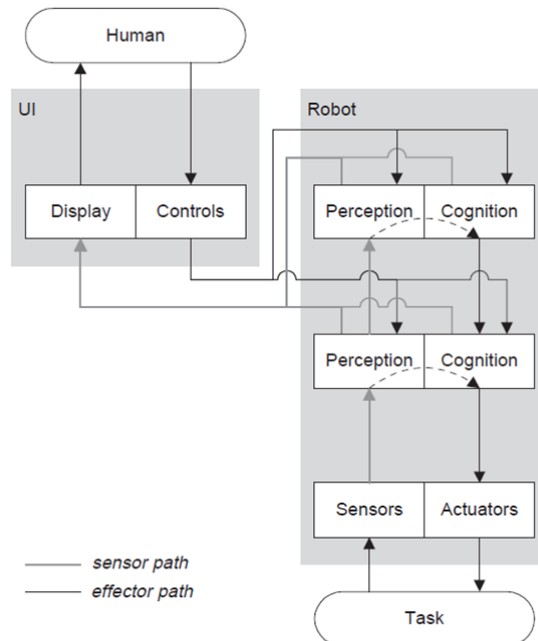

Figure 3. Fong's (2001) Collaborative Control System Model

Coactive System Model

What distinguishes joint activity from individual activity? Clark (1996) observes that "a person's processes may be very different in individual and joint actions, even when they appear identical." For example, Clark (1996) contrasts playing a musical solo versus playing a duet. 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 4. 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.
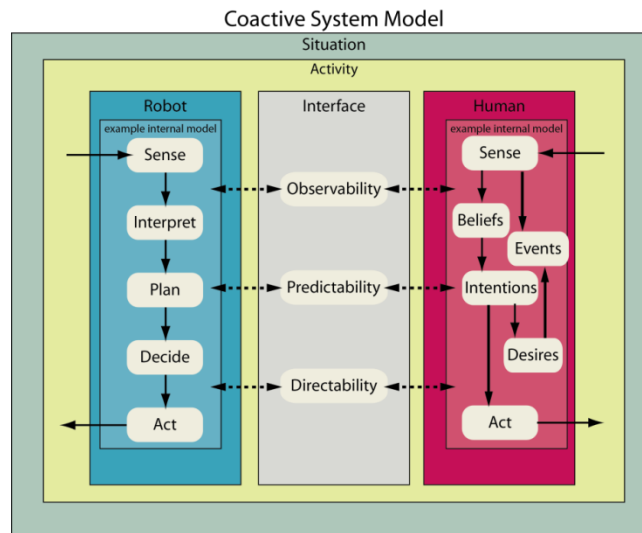

Figure 4. Coactive System Model based on observability, predictability, and directability (OPD)[4].

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. Part of the joint action ladder is attention management, which is Challenge 9 in the 10 challenges for making automation a team player (Klein et al., 2004)[5] to ensure signals are received. Though not called "observability" in the 10 challenges (Klein et al., 2004), this concept aligns with Challenge 5, revealing status and intentions, and Challenge 6, interpreting signals. It 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.

---

[4] 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.
[5] 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.

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 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.

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 limiting the human to acting on the robot and in excluding the internal processes of the human. 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 4*. This is more in line with the original interpretation of mixed-initiative interaction (Allen, Guinn, & Horvitz, 1999) than with

Michael Goodric

Deleted: Figure

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 4*, 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 4* 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.

Coactive Design Method

With an understanding of interdependence and our system model, we can now present the general method for Coactive Design, as shown in *Figure 5*. 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.

Coactive Design Method

Identification process

input:
> Traditional task analysis.
> The composition of the team and knowledge of their capabilites.
> Anticipated situation alternatives.

process:
> Identify required capacities (e.g. knowledge, information, skills, and abilities) for each task.

> Enumerate viable team role alternatives.

> Assess capacity to perform and capacity to support.

> Identify the potential interdependence relationships in the joint activity.
> > Look for required relationships based on lack of capacity.
> > Look for required relationships based on capacity that is < 100 percent reliable.
> > Look for opportunistic relationships based on capacity overlap.

> Determine the specific OPD requirements of these relationships.
> > Identify what each party needs to make observable to the others.
> > Identify behaviors that need to be predictable by others.
> > Identify the ways parties need to direct each other.

output:
> A description of a set of interdependence relationships that should be supported
> in the implemented system and the associated OPD requirements of each.

Selection and implementation process

input:
> A description of a set of interdependence relationships that should be supported
> in the implemented system and the associated OPD requirements of each.

process:
> For each relationship, decide on appropriate mechanisms to support the requirements.
> > Identify mechanisms that ensure *sufficient* support of OPD requirements.
> > Select and implement the mechanism.

output:
> A implementation of a particular mechanism designed to support a particular
> interdependence relationship and its OPD requirements.

Evaluation of change process

input:
> A implementation of a particular mechanism designed to support a particular
> interdependence relationship and its OPD requirements.

process:
> Does this implementation affect the required OPD in any other relationships?   yes

> ↓ no

> Does this implementation add, remove, or alter any interdepedence relationships?   yes

> ↓ no

output:
> Possible solution. Evaluate using traditional human factors and performance analysis.
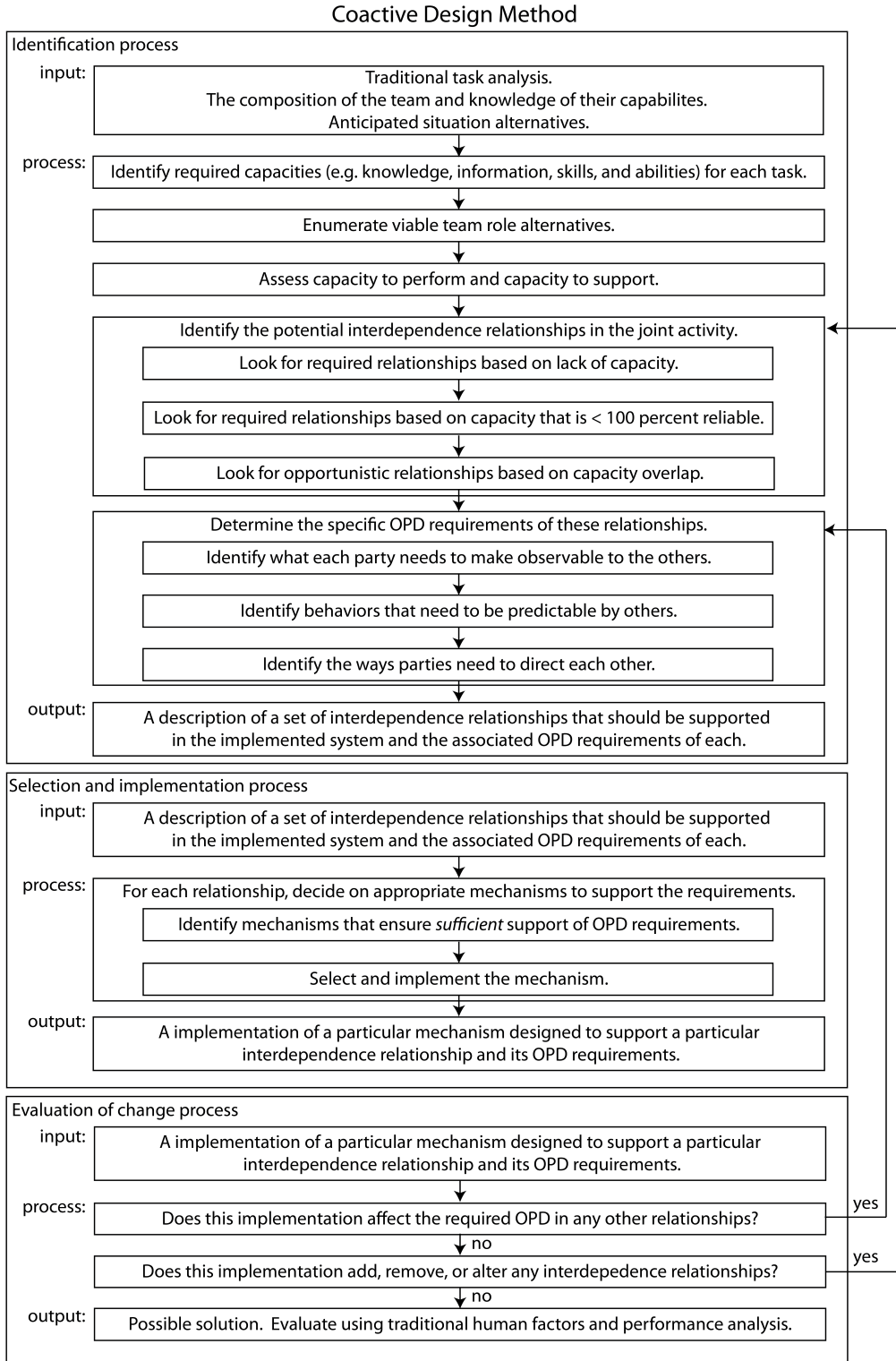
Figure 5. The Coactive Design Method

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 6*. It is similar to traditional task analysis techniques (Annett, 2003; Crandall & Klein, 2006; 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

Interdependence Analysis Table

Tasks	Hierarchical Sub-tasks	Required Capacities	Team Member Role Alternatives								
			Alternative 1				Alternative 2				
			Performer	Supporting Team Members			Performer	Supporting Team Members			
			A	B	C	D	B	C	D	A	
task	subtask	capacity									
task	subtask	capacity									
		capacity									
		capacity									
	subtask	capacity									
task	subtask	capacity									
		capacity									
	subtask	capacity									
		capacity									

Traditional hierarchical task analysis

Enumeration of viable team role alternatives

Identification of required capacties including situation awareness information, knowledge, skills, and abilities

Assessment of capacity to perform and capacity to support, as well as identification of potential interdependence relationships in the joint activity

Observability, Predictability, and Directability (OPD) requirements derive from the role alternatives the designer chooses to support, their associated interdependence relationships, and the required capacities.

Figure 6. Explanation of the different areas of the Interdependence Analysis (IA) table

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). Left-most columns of the IA table is 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 (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.

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 6*.

Assessing Capacity to Perform and Capacity to Support

After the team alternatives determined, the next step is the assessment. In order to aide future analysis, the assessment process uses a color coding scheme, as shown in *Figure 7*. The color scheme is dependent on the type of column being assessed.

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 7. 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.

Identifying Potential Interdependence Relationships

Once the assessment process is finished, the color pattern can be analyzed. *Figure 8* lists the feasible color combination based on our color scheme in *Figure 7*. Colors other than green in the "performer" column indicate some limitation of the performer, such as potential brittleness due to reliability (yellow) or hard constraints due to lack of capacity (orange). Colors other than red in the "supporting team member" columns indicate potential required (orange) or opportunistic (yellow and green) interdependence relationships between team members. This suggests 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	
Achievable	Soft	Independent operation by performer is a viable option, but assistance could improve efficiency.
	Constraint	Independent operation by performer is a viable option, but assistance could improve reliability.
	Independent	Independent operation by performer is necessary.
Potential Brittleness	Soft	Performer is < 100 percent reliable, but assistance could improve efficiency.
	Constraint	Performer is < 100 percent reliable, but assistance could improve reliability.
	Independent	Performer is < 100 percent reliable, and no assistance is possible from this team member.
Hard Constraint	Soft	Performer requires assistance, team member can provide it, and assistance can improve efficiency.
	Constraint	Performer requires assistance, team member can provide it, and assistance can improve reliability.
		Performer requires assistance, and team member can provide it.
Unachievable		Performer requires assistance, but none is possible.
		Performer can not do task.

Figure 8. 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.

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 9*. 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 9*. 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 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	
			Alternative 1	
			Performer	Supporting Team Members
			Robot	Human
Navigation	Avoid obstacles	Sense obstacles	observability	
		Interpret if obstacle is passable		
		Decide to avoid to proceed	directability	

Figure 9. 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.

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 is 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 requirements? 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.

The Evaluation of Change Process

The evaluation of change process 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.

Applying Coactive Design in the DARPA Virtual Robotics Challenge

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 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 and used the Coactive Design approach as the basis for our overall system design. 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.

The DARPA Robotics Challenge

The DARPA Robotics Challenge (DRC) was created to spur development of advanced robots that can assist humans in mitigating and recovering from future natural and man-made disasters[6]. The Virtual Robotics Challenge (VRC) was the first phase of the DRC[7]. 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. 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 10*. 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 10. 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

---

[6] http://www.darpa.mil/NewsEvents/Releases/2013/06/27.aspx (accessed on 13 JUL 2013).
[7] During Phase 2, the winners of the first phase will be given a hardware robot to be used in additional competitive activities.

to complete the tasks, and the amount of bandwidth used. After initial entries from 126[8] 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 1.

Table 1. 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

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 5, an IA table for the subtask of picking up the hose was constructed in accordance with the previous section on the identification process. The resulting IA table is shown in Figure 11.

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 _____ 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. ___was generated after a basic operator interface was developed to provide video to the operator, so ___shows that the human has the capacity. From this we can generate a set of interdependence relationships and their associated OPD requirements, as shown in *Figure 12.*

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

Michael Goodrich
Deleted: Figure

Michael Goodrich
Deleted: Figure

Michael Goodrich
Deleted: Figure

Michael Goodrich
Deleted: Figure

---

[8] 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 11. Interdependence analysis for VRC subtask of picking up the hose.

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 12*. These are used as criteria to determine if a particular mechanism we developed to support these requirements is sufficient as well as in the process of evaluating the effects of a change.

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 13*) 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.

Required Capacities	Team Member Role Alternatives				Interpretation and OPD Requirements
	1		2		
	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 12. VRC hose task interdependence relationships and OPD requirements.
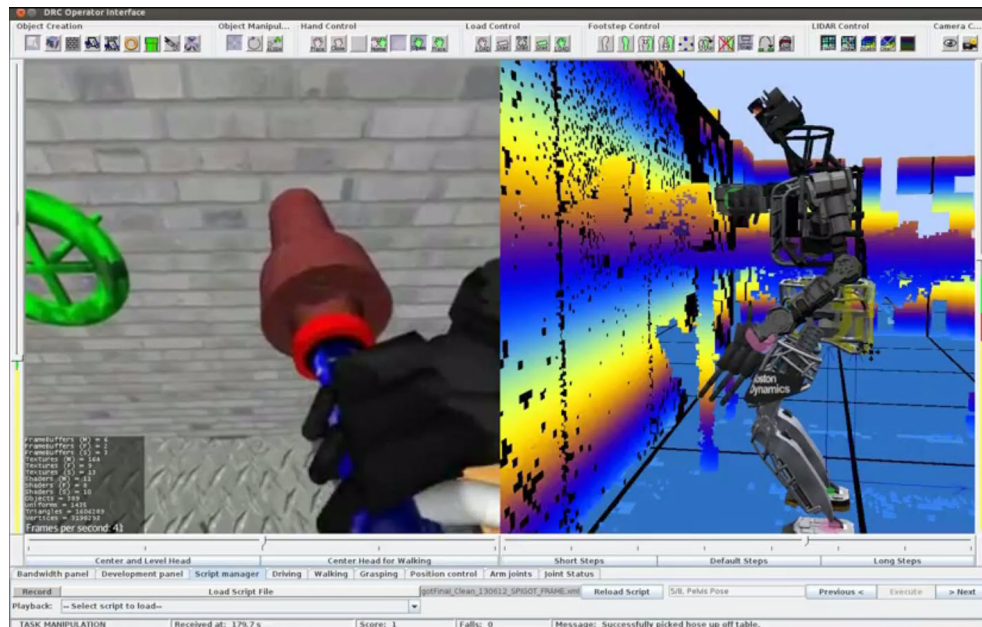

Figure 13. IHMC's VRC operator interface. The foundation of the interface is a 3D world model that is updated by state estimation from the robot. The left side provides a live video stream embedded in a 3D world model fixed to the first-person perspective. The right side is a navigable third-person view 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 14* 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.
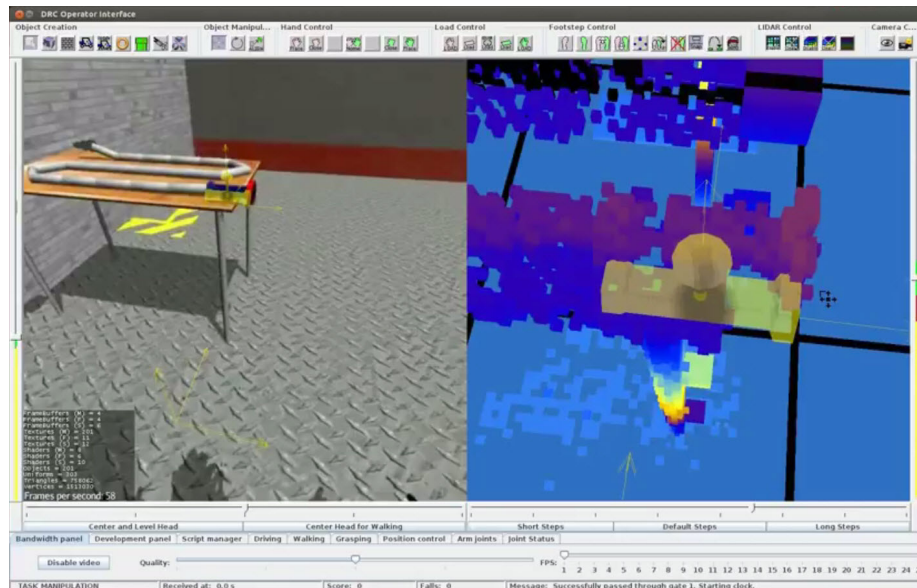

Figure 14. Hose manipulable (yellow) shown as a virtual object on both the live video (left side) and the 3D world model (right side).

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. 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 15*. 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

Michael Goodrich

Deleted: Figure

Michael Goodrich

Deleted: Figure

supporting the one percent by maintaining support for joint level control, we would have failed three out of the five runs that required it.
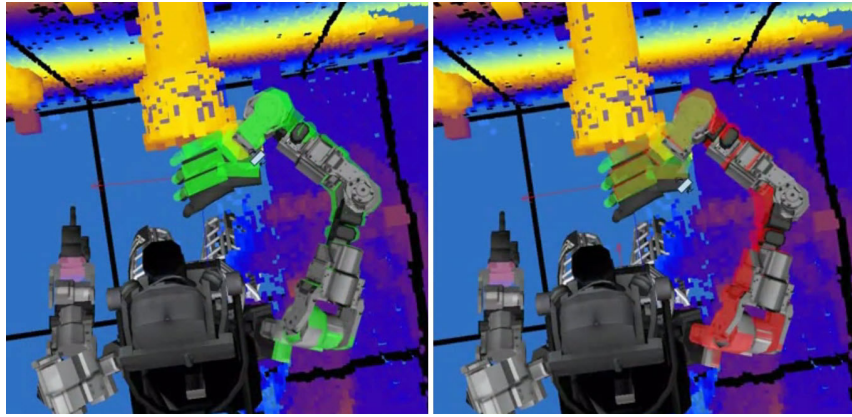

Figure 15. Inverse kinematics visuals inform the operator of valid (left side - green) and invalid (right side - red) solutions.

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 16*. This allowed the operator to position control around this point, in addition the normal control around robot joints. 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.

Michael Goodri
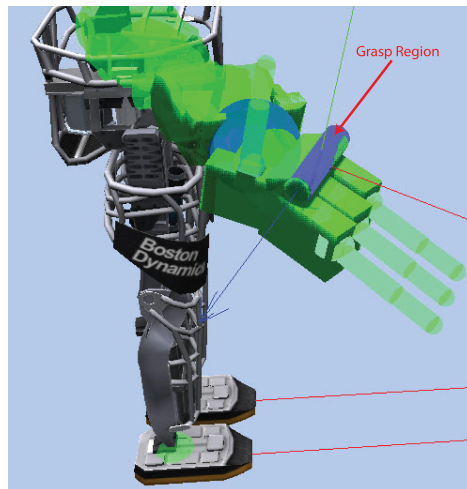
Deleted: Figure

Figure 16. 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 and a reasonable set of evaluation criteria.

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 _____ 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 table in _____ 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 table 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.

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

Michael Goodri
Deleted: Figure

Michael Goodri
Deleted: Figure

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 1 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.

Conclusion

We have introduced Coactive Design as a method to designing systems that support interdependence. We have provided definitions of what it means to be interdependent, explained why this concept is important to designers, and discussed implications of interdependence for design. Our new system model emphasizes OPD as key design guidelines for uncovering system requirements and developing detailed specifications. We presented a Coactive Design method as a starting point for designers interested in building highly interdependent systems and introduced Interdependence Analysis tables as a design and analysis tool. Finally, we demonstrated how Coactive Design was operationalized by the IHMC team during the DARRA VRC. 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.

Acknowledgements

References

Allen, J. E., Guinn, C. I., & Horvitz, E. (1999). Mixed-initiative interaction. *IEEE Intelligent Systems*, *14*(5), 14–23. 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.

Bradshaw, J. M., Feltovich, P. J., & Johnson, M. (2011). Human-agent interaction. In G. Boy (Ed.), *Handbook of Human-Machine Interaction* (pp. 293–302). Farnham, Surrey, UK: 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., … Uszok, A. (2005). KAA: Policy-based explorations of a richer model for adjustable autonomy. In *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., … 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., … 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, MA: Kluwer Academic Publishers. Retrieved from http://www.loc.gov/catdir/toc/fy036/2003040054.html

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.

Christoffersen, K., & Woods, D. D. (2002). How to make automated systems team players. *Advances in Human Performance and Cognitive Engineering Research, 2,* 1-12. Elsevier Science Ltd. doi:10.1016/S1479-3601(02)02003-9.

Clark, H H, & Brennan, S. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on Socially Shared Cognition* (127-149). Washington, D.C.: American Psychological Association.

Clark, Herbert H. (1996). *Using language*. New York, NY: Cambridge University Press. Retrieved from http://www.loc.gov/catdir/toc/cam023/95038401.html

Crandall, B., & Klein, G. (2006). *Working minds: A practitioner's guide to cognitive task analysis*. Cambridge, MA: MIT Press.

Endsley, M. R., Bolté, B., & Jones, D. G. (2003). *Designing for situation awareness: An approach to user-centered design*. Boca Raton, FL: Taylor & Francis.

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.

Fong, T. W. (2001). Collaborative control: A robot-centric model for vehicle teleoperation. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.

Hoffman, R. R., & Deal, S. V. (2008). Influencing versus informing design, part 1: A gap analysis. *IEEE Intelligent Systems*, *23*(5), 78–81.

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

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: Wiley.

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., Jonker, C. M., van Riemsdijk, B., & Sierhuis, M. (2012). Analyzing autonomy and its relation to interdependence in human-machine systems. *IEEE Intelligent Systems*, *27*(2), 43-51.

Klein, G., Feltovich, P. J., Bradshaw, J. M., & Woods, D. D. (2005). Common ground and coordination in joint activity. In W. B. Rouse & K. R. Boff (Eds.), *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. 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 *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (Vol. 3, pp. 2149–2154). doi:10.1109/IROS.2004.1389727

Larson, C. E., & LaFasto, F. M. J. (1989). Teamwork: What must go right, what can go wrong. *Sage Series in Interpersonal Communication, 10*, 150. Newbury Park, CA: SAGE Publications.

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

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

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

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

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

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. Cambridge, MA: MIT Press.

Sierhuis, M., Bradshaw, J. M., Acquisti, R., Hoof, R. Van, & Jeffers, R. (2003). Human-agent teamwork and adjustable autonomy in practice. In *Proceedings of the Seventh International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. Nara, Japan.

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. B. Cannon-Bowers & E. Salas (Eds.), *Making decisions under stress: Implications for individual and team training*(271-297). Washington D.C., American Psychological Association.

Sycara, K. (2002). Integrating agents into human teams. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 46, pp. 413–417). doi:10.1177/154193120204600342

Sycara, K., & Sukthankar, G. (2006, November). Literature Review of Teamwork Models. Pittsburgh, PA: Tech. Report CMU-RI-TR-06-50, Robotics Institute, Carnegie Mellon University.

Thibaut, J. W., & Kelley, H. H. (1959). *The social psychology of groups*. New York, NY: Wiley.

Thompson, J. D. (1967). *Organizations in action; social science bases of administrative theory*. New York, NY: McGraw-Hill.

Wyrobek, K. A., Berger, E. H., Van der Loos, H. F. M., & Salisbury, J. K. (2008). Towards a personal robotics development platform: Rationale and design of an intrinsically safe

personal robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. doi:10.1109/ROBOT.2008.4543527

Matthew Johnson, Jeffrey M. Bradshaw, and Paul J. Feltovich: The Florida Institute for Human and Machine Cognition (IHMC), Pensacola, Florida, USA. Email: mjohnson@ihmc.us, jbradshaw@ihmc.us, and pfeltovich@ihmc.us. Catholijn M. Jonker and Birna van Riemsdijk: Delft University of Technology. Email: c.m.jonker@tudelft.nl and m.b.vanriemsdijk@tudelft.nl. Maarten Sierhuis, Ejenta, and Nissan Research Center, CA , USA. Email: sierhuis@ejenta.com.