

Dynamic Execution of Temporal Plans for Temporally Fluid Human-Robot Teaming

Julie A. Shah, Brian C. Williams, and Cynthia Breazeal

MIT

Cambridge, MA 02139

julie_a_shah@csail.mit.edu, williams@mit.edu, cynthiab@media.mit.edu

Abstract

Introducing robots as teammates in medical, space, and military domains raises interesting and challenging human factors issues that do not necessarily arise in multi-robot coordination. For example, we must consider how to design robots that integrate seamlessly with human group dynamics. An essential quality of a good human partner is her ability to robustly anticipate and adapt to other team members and the environment. Robots should preserve this ability and avoid constraining their human partners' flexibility to act. This requires that the robot partner be capable of reasoning quickly online, and adapting to the humans' actions in a temporally fluid way.

This paper describes recent advances in dynamic plan execution, and argues that these advances provide a potentially powerful framework for explicitly modeling and efficiently reasoning on temporal information for human-robot interaction. We describe an executive named Chaski that enables a robot to coordinate with a human to execute a shared plan under different models of teamwork. We have applied Chaski to demonstrate teamwork using two Barrett Whole Arm Manipulators, and describe our ongoing work to demonstrate temporally fluid human-robot teaming using the Mobile-Dexterous-Social (MDS) robot.

Introduction

Collaboration between humans and robots is becoming increasingly indispensable to our work in many high-intensity domains, ranging from surgery to space exploration. We also increasingly find robots in the home, and envision a future where robots significantly enhance the quality of life for elderly and other vulnerable populations.

Introducing robot partners into these domains raises interesting and challenging human factors issues that do not necessarily arise in multi-robot coordination. For example, we must consider how to design robots that integrate seamlessly with human group dynamics. An essential quality of a good human partner is her ability to robustly anticipate and adapt to other team members and the environment. Robots should preserve this ability and avoid constraining their human partners' flexibility to act. This requires that the robot partner be capable of reasoning quickly online, and adapting to the humans' actions in a temporally fluid way.

Recent work in dynamic plan execution exhibits elements of this quality through an executive that selects and schedules activities online, dynamically in response to

a teammate's actions and other disturbances, while guaranteeing plan success. This paper describes recent advances in dynamic plan execution, and argues that these advances provide a potentially powerful framework for explicitly modeling and efficiently reasoning on temporal information for human-robot interaction.

We present formal models for two styles of human teamwork, which we name: *Equal Partners* and *Leader & Assistant*. *Equal Partners* teamwork is the fluid and coordinated interaction often associated with teammates that have trained together extensively. For example, astronauts that have trained extensively for a spacewalk exhibit *Equal Partners* teamwork. The *Leader & Assistant* style of teamwork is found in domains where there is a hierarchical relationship among team members. For example, consider a surgical nurse assisting a surgeon in the operating room. The nurse does not necessarily know what action the surgeon will take next. The nurse's role is to anticipate the surgeons' needs, offer tools and information as necessary, and above all not constrain or block the surgeon's future actions.

We describe an executive named Chaski that enables a robot to coordinate with a human to execute a shared plan under both the *Equal Partners* and *Leader & Assistant* models of teamwork. We have applied Chaski to demonstrate both models of teamwork using two Barrett Whole Arm Manipulators. Finally, we describe our ongoing work applying Chaski to demonstrate temporally fluid human-robot teaming using the Mobile-Dexterous-Social (MDS) robot.

Practical Scenario: Multi-robot Teamwork

We have successfully applied Chaski to perform multi-robot teamwork using two Barrett Arms. In this section, we present the multi-robot coordination scenario as a motivating example for the rest of the paper. Fig. 1 shows the two manipulator robots and their workspace. The robots must coordinate to remove one ball from each of the four locations in their communal workspace. Each robot also has one striped ball located in its own private workspace and must give the striped ball to the other robot using a hand-to-hand exchange. The scenario includes temporal constraints specifying the task must be completed within sixty seconds.

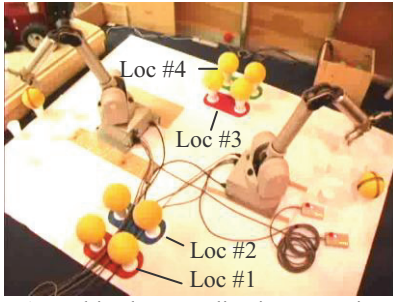


Figure 1: Multi-robot coordination scenario

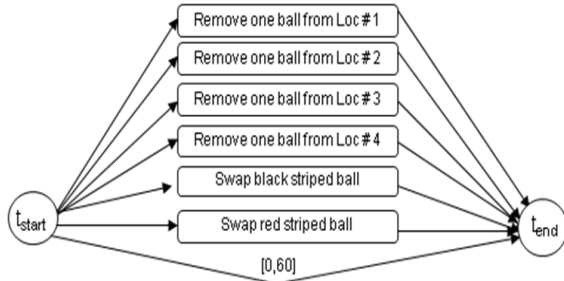


Figure 2: Plan for multi-robot coordination scenario

The plan is shown in Fig. 2. This scenario is interesting because it contains both loosely and tightly coupled interaction, and a temporal constraint on the completion of the task. Also, some activities are not a-priori allocated to a particular robot. For example, "Remove one ball from Loc #1" can be performed by either robot. Finally, the robots have heterogeneous temporal capabilities. For example, the left robot has a shorter reach distance to Loc. #1 than the right robot. As a result, removing a ball from Loc. #1 takes the left robot between 8-10 seconds and takes the right robot between 11-13 seconds.

Prior Art: Dynamic Execution of Temporal Plans

Robots must be able to execute plans while robustly anticipating and adapting to uncertainty and disturbances. For example, in the multi-robot coordination scenario, each robot must be able to adapt its plan if it accidentally drops a ball, or if its partner temporarily becomes unavailable. One way to mitigate the effect of uncertainty and disturbance is to dynamically schedule plan activities online, just before the activity is executed. This allows the agent to adapt to disturbances that have occurred prior to the activity without introducing unnecessary conservatism; this type of dynamic scheduling is called *dispatchable execution*. Dispatchable execution increases the efficiency of plan execution by introducing a *compiler* and a *dispatcher*. The compiler reduces a plan to a form that enables real-time scheduling. A temporal plan dispatcher then schedules activities online, dynamically in response to disturbances, while guaranteeing plan success. Dispatchable execution is domain independent and has been successfully applied to scheduling within the avionics

processor of commercial aircraft [Tsamardinos et al. 1998], space probes [Muscettola et al. 1998b], autonomous air vehicles [Stedl 2004], and walking robots [Hofmann et al. 2006].

Next we introduce models of temporal plans and briefly review methods for efficiently compiling and dispatching these types of plans.

Temporal Plans modeled as Simple Temporal Networks

Prior art introduces methods for efficiently compiling and dynamically dispatching temporally flexible plans modeled as Simple Temporal Networks (STNs). Agents exploit this flexible-time representation of the plan to adapt to some temporal disturbances online.

A Simple Temporal Problem (STN) is composed of a set of variables X_1, \dots, X_n , representing executable events. Events have real-valued domains and are related through binary temporal constraints. Binary constraints are of the form:

$$(X_k - X_i) \in [a_{ik}, b_{ik}]$$

Typically an activity in a temporal plan is represented with a "begin" event and an "end" event in the STN. Durations of activities are represented as *controllable* simple intervals, meaning the precise duration of the activity is within the agent's control. For example, consider the multi-robot coordination scenario. The left robot takes between 8-10 seconds to remove a ball from Loc. #1. This duration is represented in the STN as the simple interval [8,10]. This interval is controllable, meaning that the robot can control how quickly or slowly it performs this activity within the specified bounds.

The simple interval representation of temporal information enables efficient real-time scheduling of STNs. A *solution* to an STN is a schedule that assigns a time to each event such that all constraints are satisfied. [Dechter et al. 1991] describes how to check an STN for consistency, and [Muscettola et al. 1998] presents methods for compiling and dynamically scheduling STNs.

Temporal Plans modeled as Simple Temporal Networks with Uncertainty

Typically an agent only controls the timing of a subset of a plan's events; timing of the other events is controlled exogenously by nature or other agents. For example, a Mars rover can control when it starts driving to a rock; however, its precise arrival time is influenced by environmental factors. To achieve successful execution of a partially controllable plan, the scheduler must compile a dynamic execution policy that is guaranteed to be robust to these uncertainties. Since it is difficult to provide such a guarantee without any knowledge about the behavior of uncontrollable events, the scheduler exploits a model, called a *simple temporal network with uncertainty (STNU)* [Vidal 1996, Vidal and Fargier 1999], to explicitly

represent plan uncertainty by bounding the behavior of uncontrollable events.

An STNU is an extension of an STN [Dechter 1991] that distinguishes between controllable and uncontrollable durations. For example, in the multi-robot coordination scenario we specified that the left robot takes between 8-10 seconds to remove a ball from Loc. #1. By modeling this activity duration with an uncontrollable interval, we are specifying that the robot cannot control the precise duration of this activity with the specified bounds; in other words, the uncontrollable duration is free to finish any time between [8,10]. [Morris et al. 2001] introduced the dynamic controllability (DC) algorithm, showing that Simple Temporal Networks with this set-bounded representation of uncertainty can be efficiently compiled to a dispatchable form and dynamically executed.

Shortcomings of the STN and STNU Models

Simple Temporal Networks with and without Uncertainty have proven useful for important applications. However, they lack the representational power describe many problems. Specifically, STN(U)s cannot represent temporally flexible plans involving multiple agents (or resources), or choice among methods or resource allocation. As a result, we cannot use STN(U)s to model the multi-robot coordination scenario.

Recent Work: Dynamic Execution of Multi-agent Temporal Plans

In our recent work [Shah et al. 2009] we introduce a multi-agent executive named Chaski, which generalizes the state-of-the-art in dynamic plan execution by supporting just-in-time task assignment as well as scheduling.

Chaski enables an agent to dynamically update its plan in response to disturbances in the task assignment and schedule of other agents. Using the updated plan, the agent then chooses, schedules, and executes actions that are guaranteed to be temporally consistent and logically valid within the multi-agent plan. This capability provides agents (both robot and human teammates) maximal flexibility to choose task assignments, and schedule and execute activities online without the need for re-planning or plan repair. Chaski is especially useful for agents coordinating in highly uncertain environments, where near-continual plan repair results in execution delays – we see this, for example, with agents that interact with or adapt to humans.

Chaski takes as its input a multi-agent plan composed of $P=(A,V,C,L)$, where A is a set of agents, V is a set of activities, $A \rightarrow V$ is an function describing the set of feasible activities and temporal capabilities of each agent, C is a set of temporal constraints over activities, and L is a set of logical constraints (for example, resource or agent occupancy constraints). The output of Chaski is a dynamic execution policy that guarantees temporally consistent and logically valid task assignments.

Modeling the Multi-robot Coordination Scenario as a Multi-agent Temporal Plan

Consider two robots that must coordinate to perform the following four activities in the practical scenario: Remove one ball each from Loc. #1 (RB1), Loc. #2 (RB2), Loc. #3 (RB3), and Loc. #4 (RB4). The robots have heterogeneous temporal capabilities. For example, removing a ball from Loc. #1 or #2 takes the left robot 8-10 seconds and takes the right robot 11-13 seconds. We also impose the temporal constraint that all four activities must be completed within twenty seconds. Fig. 2 presents this plan described as a Disjunctive Temporal Constraint Network.

Each activity is composed of a begin event and end event. For example, "a" and "b" represent the begin and end events, respectively, for activity RB1. The amount of time each agent takes to perform the activity is represented as a disjunctive binary constraint. For example, the disjunctive constraint $L[8,10] \vee R[11,13]$ between events "a" and "b" specifies that the left robot "L" takes 8-10s to

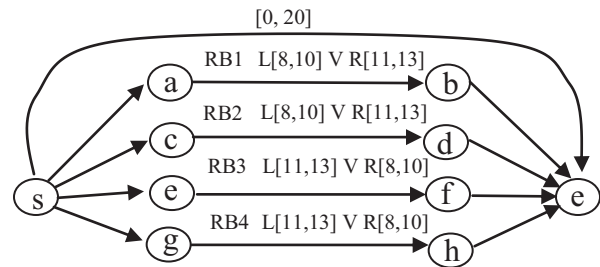


Figure 3: Multi-robot plan described as a Disjunctive Temporal Constraint Network

perform activity RB1, while the right robot "R" takes 11-13s. The execution order of the four activities is initially unspecified. The network includes ordering constraints of the form $[0, \infty]$ to specify that the activities must be executed after the epoch start event "s" and must be completed before the plan's end event "e". The temporal constraint $[0,20]$ between events "s" and "e" constrains the time available to accomplish all four activities. Note that agents do not "own" the execution of particular activity events because the plan does not specify task assignments.

In the next section we present formal models for two styles of human teamwork based on this multi-agent temporal plan representation.

Equal Partners Teamwork

Equal Partners teamwork is a fluid and coordinated style of teamwork often associated with teammates that have trained together extensively. These teammates have practiced the task so many times and in so many different ways that they are often able to "read each other's mind," and anticipate their partners' next actions. For example, astronauts that have trained extensively for a spacewalk exhibit this style of teamwork. We assume that Equal Partner teammates are not necessarily homogeneous

agents, meaning they may have dissimilar sets of feasible activities and temporal capabilities.

We model Equal Partners teamwork using a multi-agent temporal plan representation as described previously with two additional characteristics:

1. The multi-agent temporal plan involves only controllable temporal durations and assumes controllable action selection. In other words, we assume that each agent has precise control of which actions it chooses to execute as well as the timing of its actions within the specified bounds. Furthermore, each agent assumes that its partners have precise control over their action selection and timing.
2. Since the teammates have trained extensively together, we assume they have converged on a common method for updating their plan in response to their partners' actions. Furthermore, we assume that each agent has the knowledge that all agents are updating plans according to the same algorithm. In the work we describe here, we assume that each teammate updates its multi-agent temporal plan according to the execution algorithm described in [Shah et al. 2009].

The distributed execution algorithm described in [Shah et al. 2009] requires that each team member be given a copy of the shared plan. The team members then coordinate their action through communicative acts. Each agent broadcasts "status updates" to all other agents when it begins an activity, and when it finishes an activity. Studies of effective human teamwork indicate that the frequent offering of "status updates" is a very efficient and effective method for coordinating team action [Shah & Breazeal 2009]. Similarly, we are able to demonstrate efficient and fluid Equal Partners robot-robot teamwork using status updates as the only coordination mechanism.

Leader & Assistant Teamwork

The *Leader & Assistant* style of teamwork is found in domains where there is a hierarchical relationship among team members. For example, consider a surgical nurse assisting a surgeon in the operating room. The nurse's role is to anticipate the surgeon's needs, offer tools and information as necessary. The nurse does not necessarily know what action the surgeon will take next and must act so as not to constrain or block the surgeon's future actions. As with the Equal Partners model of teamwork, we assume teammates may have dissimilar sets of feasible activities and temporal capabilities.

We model Leader & Assistant teamwork using a multi-agent temporal plan representation that differs from the Equal Partners model in the following way: we model the Leader as an "uncontrollable partner." This means the Leader will choose what activities to perform and the

precise timing of its actions irrespective of the Assistant's actions. More specifically:

1. The plan models the Leader's action selection as an uncontrollable process. In other words the Assistant does not know which of the next feasible activities the Leader will choose to perform. Furthermore, the Assistant cannot influence the Leader's action selection.
2. The plan models the time the Leader takes to perform each activity as an uncontrollable temporal duration. We use this uncontrollable duration to encode that the Assistant cannot influence how long the Leader will take to perform each activity within the specified bounds.

The primary difference between the Equal Partners and Leader & Assistant models lies in the controllability of action selection and activity duration. In the next section, we describe demonstrations of multi-robot coordination. We show our models manifest these two styles of teamwork and yield interesting emerging behavior.

Demonstration of Equal Partners Multi-robot Teamwork

In this section, we describe demonstrations of Equal Partner teamwork using two Barrett Whole Arm Manipulators as robot teammates. First we narrate a *Nominal Equal Partners* demonstration where the two robots cooperatively adapt to one another to carry out the joint task described in the multi-robot coordination scenario. Next, we narrate an *Off-Nominal Equal Partners* demonstration where the robots adapt to an unexpected situation to successfully complete the task.

Fig. 4 presents snapshots of the two robots working together as Equal Partners to carry out the joint task under tight time constraints¹.

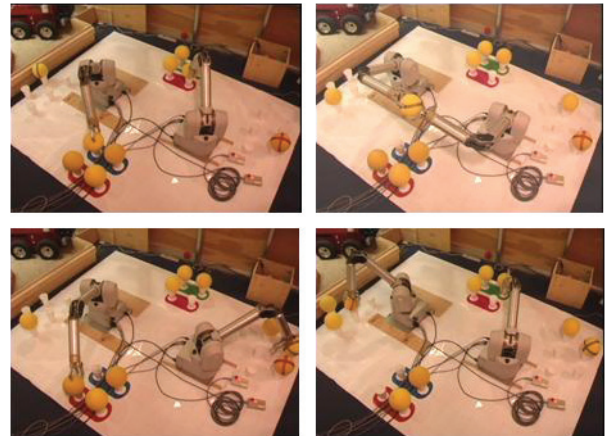


Figure 4: *Nominal Equal Partners Demonstration:* robots act as Equal Partners in performing task under tight time constraints

¹ http://people.csail.mit.edu/julie_a_shah/arm2_4x.mp4

Recall that the plan constrains the entire task to be completed within sixty seconds. This is a tight time constraint for this particular task and so the robots must coordinate efficiently: as their first action they each choose to pick up a ball from the bin closest to themselves. In Fig 4, upper left, the Left Robot picks up a ball from the blue bin in Loc. #2, and the Right Robot picks up a ball from the pink bin in Loc. #3. Next (Fig. 4 upper right) the Left Robot picks up the black-striped ball and initiates a hand-to-hand exchange. As the Right Robot puts the black-striped ball away (Fig. 4 lower left), the Left Robot opportunistically chooses and schedules the activity to pick up a ball from the red bin in Loc. #1. Next, the Right Robot picks up the red-striped ball and hands the ball to the Left Robot. Finally, the Right Robot picks up the last ball in the green bin in Loc. #4 (Fig. 4 lower right) as the Left Robot puts away the red-striped ball. The robots achieve this fluid and efficient plan execution by coordinating their actions through "status updates" as described previously.

Next we narrate a demonstration where we inject an error as the robots perform the same shared task. In particular, we highlight how the robots adapt their actions in response to this error. Fig. 5 presents snapshots of the *Off-nominal Equal Partners Demonstration*².

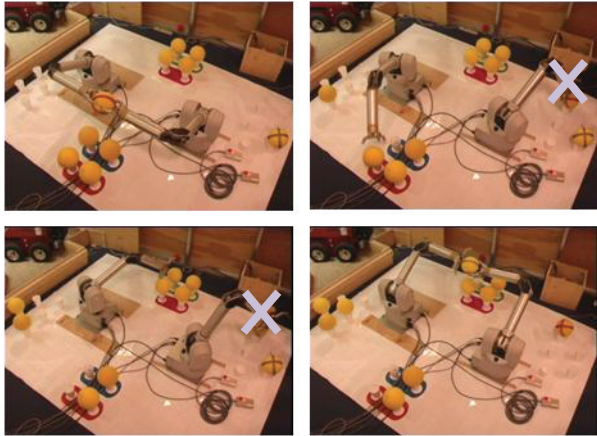


Figure 5: *Off-nominal Equal Partners Demonstration*: robots act as Equal Partners adapting to a disturbance

In this demonstration, the Left Robot initiates as hand-to-hand exchange as its first action (Fig. 5 upper left). Note that this is different than the previous demonstration, where both robots first picked up balls from the bins. The plan execution varies slightly with each demonstration since there are a number of ways to complete the task within sixty seconds and the shared plan does not a-priori specify the order of activities. The robots communicate

online to decide the who will perform which activities, as well as the order and timing of activities.

Next (Fig. 5 upper right) we inject an error into the plan as follows. We temporarily render the Right Robot inactive, thus keeping the Right Robot from successfully putting away the red-striped ball. We represent this error visually with an x-mark over the Right Robot. As a result, the Right Robot is unable to report to its partner that it successfully completed the exchange activity. The Left Robot quickly updates its plan taking into account that the activity is taking longer than expected. The Left Robot then picks up a ball from the red bin in Loc. #1 (Fig. 5 upper right) and from the green bin in Loc. #4 (Fig. 5 lower left). The Left Robot is "picking up the slack" to try to ensure the team successfully completes the plan within the time constraints. Finally, the Right Robot comes back online, exchanges the black-striped ball with its partner (Fig. 5 lower right), and the robots successfully complete the task.

Demonstration of Leader & Assistant Multi-robot Teamwork

In the previous section we described demonstrations where the two Barrett Whole Arm Manipulator robots worked together as Equal Partners to carry out a shared task. In this section, we narrate a *Leader & Assistant Demonstration*³ where the Left Robot acts as the leader and the Right Robot acts as the leader's assistant.

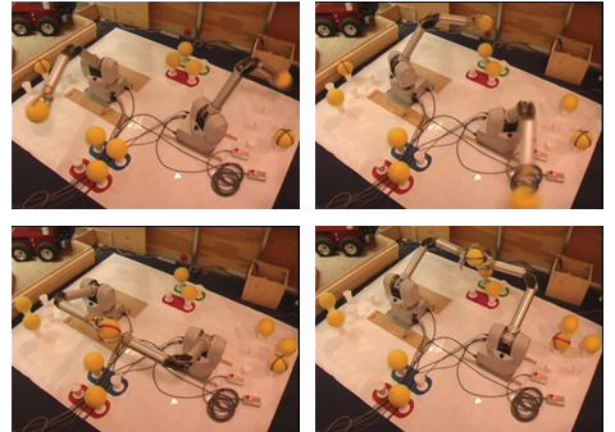


Figure 6: *Leader & Assistant Demonstration*: Left Robot is the leader and Right Robot is the assistant.

Under this model of teamwork, the Right Robot (assistant) does not know what activity the Left Robot (leader) will choose to perform next, and is not allowed to constraint the Left Robot's flexibility to act. Additionally, the Right Robot (assistant) does not know precisely how long the Left Robot (leader) will take to perform each

² http://people.csail.mit.edu/julie_a_shah/arm3_4x.mp4

³ http://people.csail.mit.edu/julie_a_shah/arm4_4x.mp4

activity, and is not allowed to act to constrain the timing of the Left Robot's actions.

Modeling the Leader's action as uncontrollable processes in this way yields interesting emerging behavior. The Right Robot (assistant) manifests a "wait and see" behavior. Early in the plan execution (Fig. 6. upper left and right) the Right Robot must wait to see what the Left Robot (leader) chooses as its next action before making its own decision. The Right Robot acts in this way to avoid "blocking" one of the Left Robot's potential next actions. Since the Right Robot (assistant) is unsure how long the Left Robot (leader) will take to perform each activity, the Right Robot acts quickly to ensure it will be ready for the Left Robot's next choice. Fig. 5 upper left and right show the Right Robot performing its activities very quickly compared to the Left Robot.

Once the robots have removed all balls from the bins, the Left Robot (leader) initiates a hand-to-hand exchange of the red-striped ball (Fig. 5 lower left). The Right Robot (assistant) has chosen its activities and timings up to this point to ensure it is ready to assist the Left Robot with this activity. Finally, the robots exchange the black-striped ball and successfully complete the task.

Future Work: Demonstrations of Human-Robot Teamwork

In this paper, we have presented two models of teamwork, and have described the behavior these models manifest through demonstrations of robot teamwork.

We argue that these models provide a potentially powerful framework for explicitly modeling and efficiently reasoning on temporal information for human-robot interaction. The next step in this ongoing work is to apply these models to demonstrate temporally fluid human-robot teaming.

In the demonstrations described in this paper, the robot teammates coordinate their actions through frequent status updates. We know from human teamwork studies that effective human teams use frequent status updates to coordinate action [Entin & Serfaty 1999, Shah & Breazeal 2009]. However, human teammates also use other types of verbal communication as well as non-verbal cues [Orasanu 1990, Stout et al. 1999, Shah & Breazeal 2009]. We are currently enhancing the multi-agent plan execution capability described in this paper to incorporate the coordination behaviors of highly effective human team members. Based on insights from human teamwork studies [Shah & Breazeal 2009], the capability will respond to a person's spoken preferences, explicit commands, and gestures (such as finger-points) in ways that are similar to interactions with another person. We are planning on validating this capability in human subject experiments where people interact with the Mobile-Dexterous-Social (MDS) robot to perform a collaborative task.

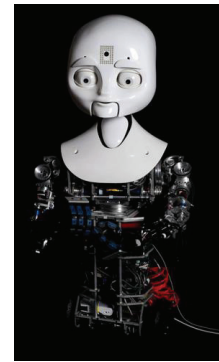


Figure 7: Mobile-Dexterous-Social (MDS) Robot, MIT Media Lab Personal Robotics Group.

References

- [Dechter, R., et al. 1991] Temporal constraint networks. *AI*, 49:61-95.
- [Entin, E., & Serfaty, D. 1999]. Adaptive team coordination. *Human Factors*, 41, 312–325.
- [Hofmann, A., Williams, B. 2006] Robust execution of temporally flexible plans for bipedal walking devices. *Proc. ICAPS-06*.
- [Morris, P., Muscettola, N., Vidal, T. 2001] Dynamic Control of plans with temporal uncertainty. In: *Proc. IJCAI-01*.
- [Muscettola, N., et al. 1998]. Reformulating temporal plans for efficient execution. *Proc. KRR-98*.
- [Muscettola, N., Nayak, P., Pell, B., Williams, B. 1998b] To boldly go where no AI system has gone before. *AI* 103(1):5-48.
- [Orasanu, J. 1990]. Shared mental models and crew decision making (CSL Report No. 46). Princeton, NJ: Princeton University, Cognitive Science Laboratory.
- [Shah, J., et al. 2009] Fast Distributed Multi-agent Plan Execution with Dynamic Task Assignment and Scheduling. *Proc. ICAPS-09*.
- [Shah, J., and Breazeal, C. 2009] An Empirical Analysis of Team Coordination Behaviors and Action Planning With Application to Human-Robot Teaming. To Appear in *Human Factors*.
- [Stedl 2004] Managing Temporal Uncertainty Under Limited Communication: A Formal Model of Tight and Loose Team Communication, S.M. Thesis, MIT.
- [Stout, R. et al. 1999] Planning, shared mental models, and coordinated performance: An empirical link is established. *Human Factors*, 41, 61–71.
- [Tsamardinos, I., et al. 1998] Fast transformation of temporal plans for efficient execution. *Proc. AAAI-98*.
- [Vidal, T., Ghallab, M. 1996] Dealing with uncertain durations in temporal constraint networks dedicated to planning. *Proc. ECAI-96*.
- [Vidal and Fargier 1999] Handling contingency in temporal constraint networks: from consistencies to controllabilities. *J. of Exp. & Th. AI*, 11:23-45.