

# Learning Temporal Plans from Observation of Human Collaborative Behavior

Sonia Chernova and Cynthia Breazeal

Personal Robots Group  
M.I.T. Media Lab  
20 Ames Street, E15-468  
Cambridge, MA 02139

## Abstract

The objective of our research effort is to enable robots to engage in complex collaborative tasks with human-robot interaction. To function as a reliable assistant or teammate, the robot must be able to adapt to the actions of its human partner and respond to temporal variations in its own and its partner's actions. Dynamic plan execution algorithms provide a fast and robust method of executing collaborative multi-robot tasks in the presence of temporal uncertainty. However, current state of the art algorithms, rely on hand-crafted plans, providing no means of generating plans for new tasks. In this paper, we outline our approach for learning a model of collaborative robot behavior by observing human-human interaction of the target task. Through statistical analysis of the recorded human behavior we extract patterns of common behavior, and use the resulting model to learn a temporal plan. The result is a learning framework that automatically produces temporal plans for use with dynamic planning that model human collaborative behavior and produce human-like behavior in the robot. In this paper, we present our current progress in the development of this learning framework.

## Introduction

The objective of our research effort is to enable robots to learn robust and temporally fluid models of collaborative behavior based on observation of human-human interaction. To reliably assist a human teammate, a robot must not only perform functionally valid actions, but must also behave in a predictable manner, be able to anticipate the actions of its partner, and adapt and respond to human actions. To achieve this level of interaction in a robust manner, the robot must be able to reason about action duration, failure recovery, and the temporal uncertainty in its own behavior and the actions of others.

Recent research in the area of dynamic plan execution has led to the development of techniques that address many of the above challenges by enabling robots to absorb some temporal disturbances at runtime through the use of a flexible-time plan representation (Brenner 2003; Lemai and Ingrand 2004; Smith et al. 2006). Such systems typically utilize

a two-step planning process composed of *task assignment* to allocate elements of the task to different teammates, and *synchronization* to enforce ordering constraints and handle concurrency. The outcome of the planning process is a Simple Temporal Network (STN) (Dechter, Meiri, and Pearl 1991), which is then dynamically executed by an *executive* that schedules the execution of task elements online as the task progresses (Muscettola, Morris, and Tsamardinos 1998). The result is a system that is able to adapt to temporal uncertainty by reassigning actions in response to changes that occur before execution of planned actions. However, events that result in re-assignment or re-synchronization due to action failure still require plan re-planning or repair. Both of these are computationally expensive processes that can significantly slow task execution for highly uncertain or dynamic tasks, making them intractable for use in tasks involving human teammates.

The recently introduced Chaski framework for dynamic plan execution (Shah, Conrad, and Williams 2009) addresses the action failure problem through just-in-time task assignment and synchronization. Given a temporal plan, the Chaski executive enables an agent to dynamically update its plan online in response to the actions of other agents and disturbances in plan execution. The result is a temporally flexible framework that enables the agent to select, schedule and execute actions that are guaranteed to be temporally consistent and logically valid within the multi-agent plan. Experimental evaluation using two robotic arms has shown the Chaski framework to be highly effective in scheduling actions in a way that is robust to variations in action duration (Shah, Conrad, and Williams 2009).

The above describes the current state-of-the-art systems. They are robust to action failures, temporal variations in action durations, and are able to respond appropriately to variations in teammate behavior and the environment. However, systems such as the Chaski executive require that a task plan be provided for the executor to execute. The plan must describe not only the actions that need to be taken and their ordering constraints, but also the temporal bounds on the execution time of each action. In all work to date these plans have been manually constructed by the human user, with temporal constraints estimated based on previously observed robot behavior. Developing a means of learning temporal plans would provide the robot with a powerful tool for

autonomously expanding its abilities through the observation of human collaborative behavior.

In this paper, we outline an algorithm for learning temporal plan representations based on observations of human collaborative task execution. In transferring collaborative behavior from humans, our goals are to enable the robot to generate behavior that is not only effective in performing the task, but also conforms to expected social norms and is predictable for the human partner. Using models of inter-human interaction obtained from analysis of recorded observations, our goal is to generate robust and temporally fluid autonomous robot behavior that closely resembles that of a human. To ensure that our probabilistic model is robust, a large volume of data will be gathered in a virtual game environment that closely resembles the real-world environment in which the robot will be tested. The resulting models of human behavior will then be used to learn a temporal plan.

The specific steps of our approach are as follows:

1. Design and deploy a multi-player online game that logs the interaction between two players as they perform a collaborative task.
2. Collect a large corpus of data recording the physical actions, text messages and gestural communication used for coordination by the human players.
3. Analyze the data and generate a statistical Plan Network model that encodes context-sensitive patterns of expected behavior (Orkin and Roy 2007)
4. Develop an algorithm that allows us to use our data corpus and the Plan Network model to generate a multi-agent temporal plan, encoded using the Reactive Model-based Programming Language (RMPL) (Ingham, Ragno, and Williams 2001)
5. Apply the Chaski framework to execute the resulting temporal plan, allowing two simulated game agents to perform the collaborative task.
6. Evaluate temporal plan execution using a physical robot in a real-world environment that closely resembles the gaming task.

In the following section we present an overview of the Chaski executive. We then present the target collaborative domain, followed by a detailed outline of each step in the development pipeline.

## Chaski

Collaborative multi-agent systems typically utilize a two-step planning process composed of task assignment, to allocate activities to each agent, and synchronization, to ensure that all activity constraints are met. The Chaski executive (Shah, Conrad, and Williams 2009) enables the execution of temporal flexible plans with just-in-time task assignment and synchronization. During online task execution, Chaski dynamically updates its temporal plan in response to the actions of other agents and disturbances in action execution, such as delay and failure. Using the updated plan, each agent then selects, schedules and executes actions that are guaranteed to be temporally consistent and logically valid

within the multi-agent collaborative plan. Unlike previous solutions, Chaski does not require expensive re-planning or plan repair, providing a fast and efficient solution for dynamic plan execution.

The input to Chaski is a temporal plan represented as  $P = (A, V, C, L)$ , where  $A$  is the set of agents,  $V$  is the set of activities,  $C$  is the set of temporal constraints over activities, and  $L$  is the set of logical constraints, such as resources. An additional function  $A \rightarrow V$  defines the set of activities each agent can perform, and their temporal constraints. Traditionally, temporal plan  $P$  that serves as the input to Chaski is manually generated ahead of time by the user. To help the user concisely encode the temporal plan, previous work has introduced the Reactive Model-based Programming Language (Ingham, Ragno, and Williams 2001; Effinger, Hofmann, and Williams 2005). RMPL is a concise task-level programming language that can be compiled to generate the complete temporal plan which is then executed by Chaski. We provide an example of a plan encoded in RMPL at the end of the paper. Using this type of plan, the Chaski executive generates a dynamic execution policy that guarantees temporally consistent and logically valid task assignments.

## Task Domain

The task domain used in our experiments is a collaborative search and retrieval task involving a human and a robot. The goal of the human-robot team is to locate the target objects and deposit them into specified bins. To encourage users to perform the desired behavior, the task will be framed as a competition, with different amounts of points awarded for depositing the various objects into the correct bins. The total score for the challenge will be based on the sum of the points awarded for object collection. Time pressure will be introduced via a pre-defined time limit.

In the virtual game scenario, two human players take on the roles of two different characters – a robot avatar and a human astronaut avatar (Figure 1). The game introduces the players to the search and retrieval task and the overall game scenario, in which the teammates must quickly collect all critical supplies and make it to their spaceship before the oxygen supplies run out. The abilities of each avatar are constrained to the real-world abilities of their respective physical embodiments. Players are randomly paired together online, and can communicate with each other through on-screen text messaging and the gestures and actions of their avatars. Before the start of the game, players are made aware that they are participating in a research study.

The game environment is designed to take advantage of the complementary abilities of the robot and human. Although the players are not given explicit instructions or roles, the environment is designed to include affordances specific to both types of players. Some objects are accessible only to the robot through the use of locked, motorized boxes that can only be opened by special RFID “keys”, while other objects are more easily detected by the robot’s Organic Compound Scanner than the human eye. The human player, on the other hand, will have greater navigational speed and dexterity, and will be better able to reach objects



Figure 1: Screenshot of the online game environment.

in cluttered environments. Successful completion of the task will rely on communication and collaboration between both teammates.

On online version of this game will be used to gather a data corpus consisting of hundreds of interactions. This dataset will be used to learn models for autonomous collaborative robot behavior that will then be evaluated using a physical robotic platform. For the evaluation, the simulated game environment will be recreated as closely as possible in the real world. The robot used for this task will be the Mobile-Dexterous-Social (MDS) robotic platform, which combines a mobile base with an anthropomorphic upper body (Figure 2). The MDS robot is equipped with a sophisticated biologically-inspired vision system that supports animate vision for shared attention to visually communicate the robot's intentions to human observers. The auditory inputs support a microphone array for sound localization, as well as a dedicated speech channel via a wearable microphone for speech recognition. Two dexterous hands provide capability to grasp and lift objects.

Due to the complexity of the search and retrieval task, a high precision offboard Vicon MX camera system will be used to supplement the robot's onboard sensors. The Vicon system will be used to track the position of the robot, human, and objects in the environment in real time using lightweight reflective markers attached to object surfaces. This tracking system will enable the robot to have a greater degree of environmental awareness that is similar to that of a human. The human teammate will be fitted with uniquely marked hat and gloves to enable the system to accurately identify the direction of the person's gaze and gestures. This information will be critical for inferring the contextual meaning of the user's utterances.

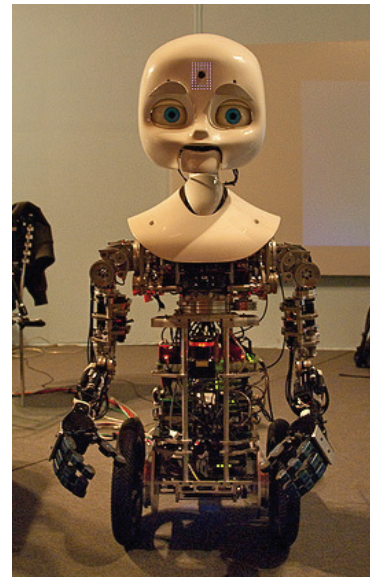


Figure 2: The Mobile-Dexterous-Social (MDS) robot.

## Human Performance Corpus

A large body of data capturing a diverse set of interactions with many different people is required in order to create a robust statistical model of typical collaborative human behavior. Acquiring this data in real-world trials would be prohibitively expensive, requiring many hours to recruit, train and record study participants. Instead, our approach will take advantage of a simulated world environment that closely resembles to real-world space in which the evaluation will be conducted.

Our approach is motivated by the recently developed "Restaurant Game" (Orkin and Roy 2007; 2009). This minimal investment multiplayer online (MIMO) game enabled users to log in to a virtual environment and take the role of one of two characters, a customer or a waiter, at a restaurant. Players were randomly paired with another online player, and could interact freely with each other and objects in the environment. In addition to standard game controls, the users could maintain dialog with each other, and other simulated characters, by typing freeform text. Logs of over 5,000 games were used by the authors to analyze this interactive human behavior and acquire contextualized models of language and behavior for collaborative activities.

Similar to the design of the Restaurant Game, our simulated world will enable users to log in and play the role of either the robot or the human avatar. The online environment will closely resemble its real-world counterpart in which the final evaluation will take place. The set of available actions and their speed will closely approximate real-world actions for each player. The human avatar will be able to move to any location in the space, as well as pick up, carry, drop, and pass objects. The robot avatar will be able to perform a similar set of actions, with the addition of being able to unlock boxes and use specialized sensors. The robot's movements will be slower than those of the person, and its actions will

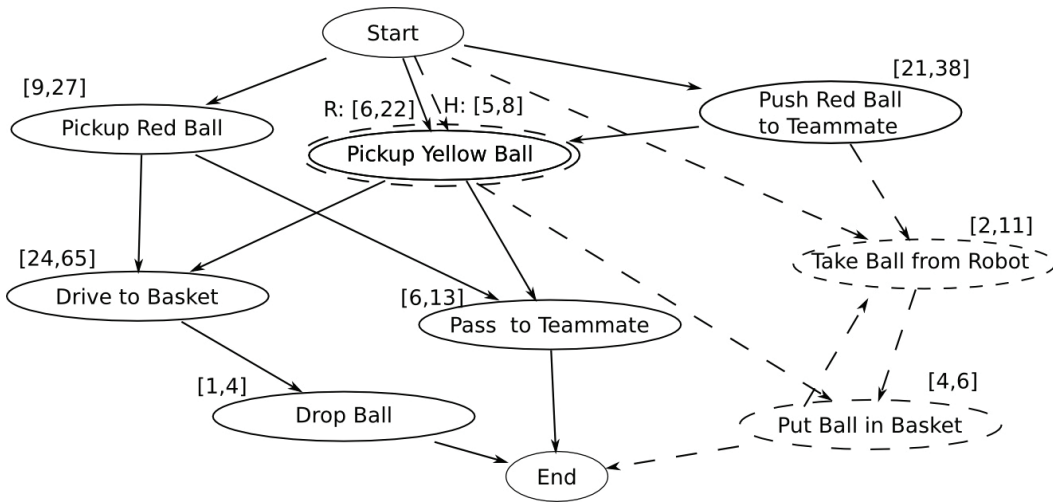


Figure 3: An example Plan Network representing a ball retrieval task in which a red and yellow ball are collected into a basket. Solid lines represent the robot’s actions and dashed lines represent human actions. Numerical ranges represent the temporal bounds on the duration of each action.

have a non-zero probability of failure. In addition to controlling an avatar, each player will be able to communicate with his or her teammate by typing freeform text, and by using the mouse to highlight objects of interest. The purpose of the mouse interface is to simulate gesture-based communication, such as pointing, which will be present in the real-world interaction.

Logs recorded during game execution will record changes to world state that occur at each timestep, such as player movements, spoken phrases and actions taken upon the environment. Following the completion of each game, activity recognition analysis will be performed to abstract individual movements of the players into high-level action categories (e.g. "GoTo") and descriptors (e.g. "door"). In the following section we describe how the data will be used to build a statistical model of typical collaborative behavior.

### Learning Expected Patterns of Behavior

Data recorded in the simulated environment will provide examples of a wide variety of human behavior. Examples will include many valid but different action orderings and dialog interactions, as well as a large variety of extraneous actions people may have taken while exploring, becoming familiar with the environment, or just for fun. Our intention in structuring the task as a timed challenge is to encourage all users to have the same goals and to produce collaborative behavior aimed at achieving those goals. We therefore expect that analysis of the recorded data will enable us to bring out the main behavioral trends that capture the core of the interactive behavior.

To model what a “typical” interaction might look like, we will take advantage of the Plan Network algorithm developed by Orkin and Roy for the Restaurant Game (Orkin and Roy 2007). A Plan Network is a statistical model that encodes context-sensitive expected patterns of behavior and language. Given a large corpus of data, a Plan Network can

not only provides a graphical visualization of all possible actions taken, but can also be used to filter out aberrant behaviors that occurred infrequently and were not likely to be a key component of the collaboration.

Figure 3 presents a Plan Network for a simple example task in which a red and yellow ball must be retrieved and placed into a bin. Each node of the network represents an action, and vectors connecting the actions show the typical ordering of actions observed. Data for both the robot and human player is displayed – solid lines represent actions of the robot and dashed lines those of the human. Additionally, our game logs provide us with the ability to estimate expected action duration, and each node in the Plan Network is labeled with the observed temporal bounds for that action in seconds ([min, max]). When the same action can be performed by both players, temporal information is maintained separately for the robot (R) and human (H) (e.g. Pickup Yellow Ball). By analyzing the network, we see that both players can pick up and handle the yellow ball, but only the robot is able to initially handle (pick up or push) the red ball, although it can then choose to pass it to the human. The robot’s actions are less predictable than those of a human, and have a higher temporal variance.

In addition to providing a model of typical behavior, the Plan Network can be used during task execution to detect unusual behaviors performed by a user that are outside the expected norms. Using likelihood estimates, the model is able to score the typicality of a given behavior by evaluating how likely this behavior is to be generated by an agent controlled by the learned model. Atypical behaviors not directly related to the task are highly likely to occur in many instances of human-robot interaction. Providing the robot with an ability to respond to such events appropriately may have powerful impact on how the person perceives the robot and on the overall success of the interaction. An interesting direction for future work is to examine how the robot

should behave when its teammate goes “off-script”; politely reminding the person to get back to work, or simply continuing to perform the task alone are two possible options.

### Learning a Temporal Plan

The Plan Network described in the previous section provides us with a model representing a wide range of possible behavioral progressions that form the basis of the collaborative task. Using the frequency count to select among transitions, the system could use the Plan Network to enable the robot to autonomously reproduce some of the more commonly encountered, and likely logical, behavioral sequences. However, this behavior selection mechanism lacks the ability to reason about the actions of the other player, the overall goals of the interaction and the temporal constraints of the system. The robot’s actions are therefore likely to be semantically invalid and are not guaranteed to complete all elements of the task (e.g. some of the objects may not be picked up by either player).

Our future work will therefore focus on the development of an algorithm for automatically generating a temporal plan representation of the task that can be used by the Chaski executive to perform dynamic plan execution. This approach will result in robust, temporally fluid and semantically valid task execution that, we predict, will also conform to the social and expected norms of human behavior. To learn the temporal plan, which will be represented in the Reactive Model-based Programming Language, we will leverage the Plan Network structure. RMPL is an expressive programming language that contains constructs for concurrency, sequencing, iteration, preemption, conditional branching and probabilistic transitions. Table 1 presents a manually coded RMPL representation for the ball retrieval Plan Network from Figure 3. Even for a relatively simple task, the resulting RMLP representation is complex, further motivating the need for the automated plan generation methods that will be the focus of our future research.

### Evaluation in Real-World Environment

In the final stage of the project we will evaluate the performance of the learned collaborative behavior in a complex, real-world search and retrieval task that closely resembles the online gaming environment. A broad range of adult study participants will be recruited to perform the task in collaboration with an autonomous MDS robot controlled through dynamic plan execution. Activity recognition of the person’s actions, together with behavioral cues, such as speech and gestures, will be used to adapt the robot’s behavior to that of the human. The robot’s ability to perform human-like gestures, gaze, facial expressions, and speech will further aid in communication between the teammates.

All teammate interactions within the real-world environment will be logged, and users will be asked to complete an additional survey describing their experiences. Using this data we will evaluate the algorithm and its variations for a wide range of performance metrics, including:

- team task performance
- how well the robot responded to human actions

```

BallRetrievalTask() [30,120] = {
  parallel (
    choose (
      sequence (
        R: PickupRedBall [9,27],
        choose (
          sequence (
            R: DriveToBasket [24,65],
            DropBall [1.4]
          ),
          parallel (
            R: PassToTeammate [6,13],
            sequence (
              H: TakeBallFromTeammate [2,11],
              H: PutBallInBasket [4,6]
            )
          )
        )
      )
    )
    parallel (
      R: PushRedBallToTeammate [21,38],
      sequence (
        H: TakeBallFromTeammate [2,11],
        H: PutBallInBasket [4,6]
      )
    )
  )
  sequence (
    H: PickupYellowBall [5,8],
    H: PutBallInBasket [4,6]
  )
)
if ( yellowBallNotRetrieved ) thennext (
  choose (
    sequence (
      R: PickupYellowBall [6,22],
      choose (
        sequence (
          R: DriveToBasket [24,65],
          R: DropBall [1.4]
        ),
        parallel (
          R: PassToTeammate [6,13],
          sequence (
            H: TakeBallFromTeammate [2,11],
            H: PutBallInBasket [4,6]
          )
        )
      )
    )
  )
  sequence (
    H: PickupYellowBall [5,8],
    H: PutBallInBasket [4,6]
  )
)
}

```

Table 1: The ball retrieval task represented in the Reactive Model-based Programming Language.

- how often the robot's actions required adaptation or correction on behalf of the human teammate
- how well the robot's behavior adhered to human expectations
- how well the human teammates understood what the robot was doing and why

## Conclusion

In this paper, we outlined an ambitious project for learning complex temporal plans of collaborative human-robot behavior based on observations of human-human collaboration in a simulated environment. Once complete, this project will provide a novel and powerful framework for generating autonomous robot behavior for human-robot interaction that takes into account temporal variations and constraints.

## References

- Brenner, M. 2003. Multiagent planning with partially ordered temporal plans. In *International Conference on AI Planning and Scheduling*.
- Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1-3):61–95.
- Effinger, R.; Hofmann, A.; and Williams, B. C. 2005. Progress towards task-level collaboration between astronauts and their robotic assistants. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (ISAIRAS-05)*.
- Ingham, M.; Ragno, R.; and Williams, B. 2001. A reactive model-based programming language for robotic space explorers. In *in: Proceedings of ISAIRAS-01*.
- Lemai, S., and Ingrand, F. 2004. Interleaving temporal planning and execution in robotics domains. In *AAAI'04: Proceedings of the 19th national conference on Artificial intelligence*, 617–622. AAAI Press / The MIT Press.
- Muscettola, N.; Morris, P.; and Tsamardinos, I. 1998. Reformulating temporal plans for efficient execution. In *In Principles of Knowledge Representation and Reasoning*.
- Orkin, J., and Roy, D. 2007. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development* 3(1):39–60.
- Orkin, J., and Roy, D. 2009. Automatic learning and generation of social behavior from collective human gameplay. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 385–392. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Shah, J. A.; Conrad, P. R.; and Williams, B. C. 2009. Fast distributed multi-agent plan execution with dynamic task assignment and scheduling. In *ICAPS*.
- Smith, S.; Gallagher, A. T.; Zimmerman, T. L.; Barbulescu, L.; and Rubinstein, Z. 2006. Multi-agent management of joint schedules. In *Proceedings of the 2006 AAAI Spring Symposium on Distributed Plan and Schedule Management*. AAAI Press.