Timing in Multimodal Turn-Taking Interactions: Control and Analysis Using Timed Petri Nets

Crystal Chao, Andrea L. Thomaz
Georgia Institute of Technology

Turn-taking interactions with humans are multimodal and reciprocal in nature. In addition, the timing of actions is of great importance, as it influences both social and task strategies. To enable the precise control and analysis of timed discrete events for a robot, we develop a system for multimodal collaboration based on a timed Petri net (TPN) representation. We also argue for action interruptions in reciprocal interaction and describe its implementation within our system. Using the system, our autonomously operating humanoid robot Simon collaborates with humans through both speech and physical action to solve the Towers of Hanoi, during which the human and the robot take turns manipulating objects in a shared physical workspace. We hypothesize that action interruptions have a positive impact on turn-taking and evaluate this in the Towers of Hanoi domain through two experimental methods. One is a between-groups user study with 16 participants. The other is a simulation experiment using 200 simulated users of varying speed, initiative, compliance, and correctness. In these experiments, action interruptions are either present or absent in the system. Our collective results show that action interruptions lead to increased task efficiency through increased user initiative, improved interaction balance, and higher sense of fluency. In arriving at these results, we demonstrate how these evaluation methods can be highly complementary in the analysis of interaction dynamics.

Keywords: Turn-taking, engagement, reciprocity, timing, multimodality, timed Petri net, architecture, simulation, collaboration, human-robot interaction

1. Introduction

Advances in robotics have made it plausible that robots will enter our homes, schools, hospitals, and workplaces in the near future. As robots become increasingly capable machines, the reality of their communicative limitations becomes more pressing. If we want robots to engage effectively with humans on a daily basis in service applications or in collaborative work scenarios, then it will become increasingly important for them to achieve the type of interaction fluency that comes naturally between humans.

Timing is a factor of great importance to any domain where task efficiency is a concern, but timing can also have a significant impact on interaction fluency. People gradually synchronize the timing of communicative behaviors to interaction partners over time (Burgoon, Stern, & Dillman, 1995). Imbalances in the amount of time taken by each interaction partner lead to perceptions of

Journal of Human-Robot Interaction, Vol. 1, No. 1, 2012, Pages 4-25. DOI 10.5898/JHRI.1.1.Chao

power; for example, speaking quickly and over shorter durations than one's partner can convey lower dominance over the conversational floor. Altering the balance of control affects the decisions that people make in a social situation, which can then affect the outcome of a task.

In this work, we assume that HRI systems that control robots achieve appropriate semantics and perform actions in the correct order. That is, robots can be programmed to make sequential plans and select the correct response in a dialogue. In addition to this, we would like for robots to be able to manage how they time their actions. Usually, the default timing for completing an action is "immediately." For the reasons listed previously, though, this may not always be the appropriate decision. Managing timing means having an understanding of the impact that timing changes can have on interactions, as well as having a control system that enables the manipulation of timing changes with enough flexibility.

In this paper, we describe a system that is intended for the control and analysis of timing in multimodal reciprocal interactions. *Reciprocal* describes the robot's human-centered social tendency, including motivation to engage users and maintain balanced turn-taking behaviors. *Multimodal* describes the nature of social actions, which leverage the multiple channels of speech, gaze, gesture, and other instrumental body motions to perform an intricate dance with the interaction partner. *Turn-taking* is a phenomenon that arises in the presence of bottlenecking resources, such as shared physical space or the speaking floor. When humans engage in multimodal reciprocal behavior, the result is fluent and seamless turn-taking of shared resources. Our system architecture is based on the timed Petri net (TPN), which provides a formal semantics well suited for the type of control needed for multimodal reciprocal interactions, including turn-taking. Section 3 provides an overview of the system.

As one would expect, a system that attempts to do of all this is can be complex. When such a system needs to scale to incorporate novel behaviors or additional modules, it can be difficult to evaluate how various factors and their combinations contribute to overall interaction dynamics. We thus leverage TPN simulation in order to provide such factored characterizations of the system. In Section 5.4, we describe how TPN simulation can be used as a technique to analyze HRI systems dynamics.

To demonstrate the benefits of our system in more detail, we also conduct a focused study of the effects of a particular system extension, action interruptions, on turn-taking dynamics. We describe the semantics and implementation of such action interruptions in Section 4. We describe our evaluation methodologies for the extension in Section 5, which include a traditional between-groups user study with 16 human subjects and a simulation experiment with 200 simulated users. By analyzing the simulation data in conjunction with the user study data, we discuss in Section 6 how we are able to achieve better understanding of the effect of action interruptions on system and interaction dynamics.

2. Related Work

This work continues a growing body of research on engagement and turn-taking. Early work on developing architectures to manage this problem considers how nonverbal cues used by virtual agents on a screen can affect perception of lifelikeness (Cassell & Thorisson, 1999). More recent work on engagement with virtual agents uses more elaborate turn-taking models and supports multiparty conversation (Bohus & Horvitz, 2010). Research in spoken dialog systems also attempts to control the timing of turn-taking over the single modality of speech (Raux & Eskenazi, 2009). Although some results on cue usage in unembodied systems can generalize to robots, the timing of controlling actions on embodied machines differs substantially from that of virtual systems. Robots require time to move through physical space, and additionally, they must negotiate resources such as shared space and objects through turn-taking with human collaborators. Such bottlenecks arising due to

embodiment are not an issue in virtual agent communication.

In robotics, there has been study of how speaker-listener roles can be strongly shaped by controlling the single modality of eye gaze (Mutlu, Shiwa, Ishiguro, & Hagita, 2009). There has also been related work on conversational engagement. Rich et al. introduced the concept of detecting connection events, which are interaction patterns found in analysis of human data (Rich, Ponsler, Holroyd, & Sidner, 2010). Holroyd et al. subsequently showed how a robot could generate these to maintain engagement (A. Holroyd, Rich, Sidner, & Ponsler, 2011). Engagement and floor exchange are very relevant topics to the timing of turn-taking. Our system builds on prior work on conversational engagement and single-modality turn-taking by creating a framework that supports turn-taking in combinations of arbitrary modalities.

Our own investigations in turn-taking focus on modeling the relationship between timing and context, which serve as motivation for the novel system and analyses presented in this paper. In our previous Wizard of Oz data collection study (Chao, Lee, Begum, & Thomaz, 2011), participants played the imitation game "Simon says" with our humanoid robot Simon. From analysis of this data, we defined a principle of minimum necessary information (MNI) that describes how information that is redundant across multiple modalities affects human response times. That is, the end of the MNI is a critical reference point for analyzing response timing. This idea led us to specify an interface based on information flow between turn-taking and context models, and also to present a preliminary implementation of action interruptions based on MNI by extending a finite state machine (Thomaz & Chao, 2011).

We continue the previous work here through a new system based on timed Petri nets described in Section 3, a more developed interruption representation in Section 4, and an extensive evaluation in Section 5. Our architecture represents an ongoing effort to encode human turn-taking principles into a robot control system. As more rules get incorporated into the system, the system becomes increasingly complex. As we continue to develop the architecture described in Section 3, we strive to model and implement turn-taking in ways that are natural to understand and analyze, and thus scale well to additional rules.

The domain of inquiry described in Section 5.1 concerns a collaborative manipulation task, which we believe is interesting because task efficiency can be a metric for fluency. This metric has been used in previous research on human-robot teamwork, although the collaborative approaches can be diverse. Hoffman and Breazeal have used Bratman's notion of shared cooperative activity (Bratman, 1992) to develop a system that meshes human and robot subplans (Hoffman & Breazeal, 2004). This strategy is the most similar to our robot's planner for this domain. But there are other ways that robots can collaborate with humans. One system enabled a robot to play an assistive role by handing construction pieces over based on the human's eye gaze (Sakita, Ogawara, Murakami, Kawamura, & Ikeuchi, 2004). In this context, the robot is subservient to the human's intentions. In contrast, the Chaski system is a task-level executive scheduler that assigns actions to both the human and the robot by minimizing idle time (Shah, Wiken, Williams, & Breazeal, 2011). Schedulers are excellent for generating theoretically optimal plans, but a potential drawback is that reducing human control over forming task plans can lead to poorer mental models, engagement, and execution fluency.

In the larger scheme of reciprocal interactions, we hope to encapsulate fluency in ways that are not tied to particular domains, such as the completion time of a particular task. There is some previous robotics research that attempts to address timing and pacing of interactions outside of task contexts. One example is work on the Keepon robot that uses the idea of rhythmic synchrony in interactions with children (Kozima, Michalowski, & Nakagawa, 2009). Another is a percussionist robot that takes turns with human musicians during improvisation (Weinberg & Blosser, 2009). Turn-taking has also been investigated as an emergent phenomenon with a drumming robot (Kose-

Bagci, Dautenhan, & Nehaniv, 2008). All of these robots still interact in a musical context, in which musical rhythms provide high amounts of structure for the interaction. This prior work also does not seek to provide any evaluation or metrics for interaction dynamics, one of the goals of our research.

3. System Description

This section describes our techniques for modeling and control of multimodal turn-taking interactions using a timed Petri net. The literature on Petri nets is highly diverse and includes many variants, each with their own approaches to typing and timing. Petri nets have historically been used for modeling, but have not been as commonly used for control. Previous work on robot control using Petri nets include applications in assembly (Zhang, 1989) and manufacturing (Cao & Anderson, 1993). More recently, they have been used as supervisors for multi-robot control in a robot soccer domain (Barrett, 2010; Lacerda & Lima, 2011). The engagement behavior generator of (A. G. Holroyd, 2011) also makes use of Petri nets, but with a different approach; Holroyd's Petri nets are dynamically generated and executed for the realization of Behavior Markup Language (BML), rather than serving as a persistent control system.

The formalism we detail here integrates various Petri net modeling techniques that specifically support the control of multimodal reciprocal human-robot interactions. We find that Petri nets, and specifically TPNs, are an intuitive and elegant representation for developing autonomous controllers for HRI turn-taking scenarios. In the remainder of this section we describe the formalism and its application to turn-taking control.

3.1 Discrete Events

There are various ways to represent the types of discrete events that occur in a multimodal dialogue. From a transcript logged from system execution, one can produce a bar visualization depicting the alignment of the events. This format can be helpful for annotating, in conjunction with data playback, and could look something like Figure 1.

In the figure, the beginning and end of each segment represents a discrete event that is important to the interaction — that is, a state change that potentially affects a system decision. Hence, what we are concerned with is the specification of a Discrete Event Dynamic System (DEDS), which describes the potentially concurrent or asynchronous alignments of important event chains throughout a system execution. A DEDS can be expressed as a Petri net, which provides a useful set of semantics shared for both control and analysis. A more detailed review of Petri nets can be found in (Murata, 1989).

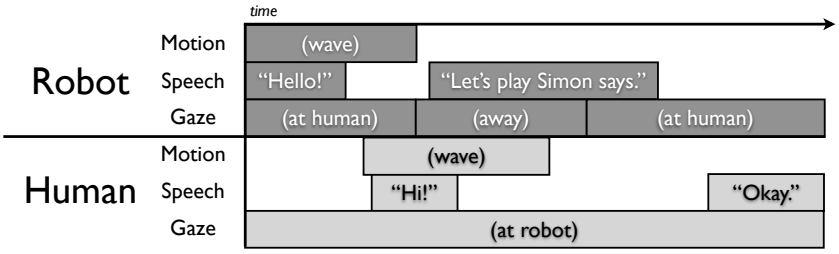


Figure 1. An example of event alignment for a multimodal interaction.

A Petri net, also called a Place/Transition (P/T) net, is a bipartite multigraph comprising four types of primitives: places, transitions, directed arcs, and tokens. Control is transferred through the

movement of tokens throughout the graph. In general, a Petri net is formally defined as a 5-tuple $N = (P, T, I, O, M_0)$, where:

- $P = \{p_1, p_2, ..., p_x\}$ is a finite set of places,
- $T = \{t_1, t_2, ..., t_y\}$ is a finite set of transitions, where $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$,
- $I : P \times T$ is the input function directing incoming arcs to transitions from places,
- $O : T \times P$ is the output function directing outgoing arcs from transitions to places, and
- M_0 is an initial marking.

Places contain a nonnegative integer number of tokens, which can represent any kind of resource. This could be a queue of parameters to be processed, a discrete variable, or a shared tool. In general, the state of a Petri net is implicitly represented through the marking $M : P \rightarrow \mathbb{N}$, the number of tokens in each place. In our system, tokens are typed objects mapped onto a value; this variant is sometimes referred to as a colored Petri net. Figure 2 shows examples of tokens having the types of Animation, SpeechAct, and Vec3.

In addition to the above general Petri net semantics, the firing mechanics for our system are as follows:

- A token $(k : \sigma) \rightarrow (v : \sigma)$ is parametrized by type σ and has value v of that type.
- A place $(p : \sigma) = \{k_1 : \sigma, k_2 : \sigma, ..., k_z : \sigma\}$ is parametrized by type σ, and contains a list of same-typed tokens.
- A transition $t = \{\mathcal{G}(I), \mathcal{F}(M, I, O)\}$ is controlled by a guard function $\mathcal{G}(I)$ and a firing function $\mathcal{F}(M, I, O)$.
- A guard function $\mathcal{G}(I) \rightarrow \{0, 1\}$ is an indicator function describing the logic for enabling transition t as a function of the inputs of t. An enabled transition executes the firing function until the transition is no longer enabled.
- A firing function $\mathcal{F}(M, I, O) \rightarrow M'$ takes as input the current graph marking M and produces new marking M' by removing tokens from the inputs of t and assigning tokens to any of its outputs, following type rules. The transition is considered to fire whenever a new marking is produced.

Typical guard functions in the system are AND-logic and OR-logic expressions, but any boolean expression is possible. Our system also uses the common addition of inhibitor input arcs, which allow places with tokens to prevent transitions from enabling.

Petri nets support a natural visualization system for graph primitives. Places are drawn as circles, transitions as rectangles, directed arcs as arrows, and tokens as small filled circles inside of places. Inhibitor arcs are drawn with a circular endpoint. Examples of this visual scheme can be seen in Figures 2, 4, 7, and 9.

3.2 Timing

Timing control and analysis is made possible with the following additional components. For a more extensive overview of the different kinds of TPNs, see (Wang, 1998).

- The system clock $C(i, \tau) \rightarrow \tau'$ determines how the current time τ updates to the new time τ' at each cycle i.

- A transition $t \stackrel{+}{=} \{\delta_e(), \delta_f(I)\}$ is additionally associated with an enabling delay function and a firing delay function.
- An enabling delay function $\delta_e() \rightarrow d_e$ calculates the delay d_e before the transition is enabled from the time that the guard function evaluates to true.
- A firing delay function $\delta_f(I) \rightarrow d_f$ calculates the expected delay d_f after the time when the transition is enabled but before the transition fires.

The clock module determines the rate at which the system is executed, the needs of which may vary depending on the application. A clock that updates τ faster than real-time is useful for simulations. The delay functions in our system are varied in structure and include immediate timers, deterministic timers, and stochastic timers (e.g. following a Gaussian distribution). An example application of injecting a timer into a control sequence is waiting in a system state for a certain duration before proceeding. In the Towers of Hanoi domain, there is a 1-second delay between a plan bottleneck and a robot verbal request, to allow the human a window of opportunity to take action before being bothered by the robot.

Timing history is tracked in a distributed manner in our system by associating histories of timing intervals $[\tau_\alpha, \tau_\beta)$ with certain graph primitives, defined as follows:

- For a place p, τ_α is recorded when $|p|_i = 0 \rightarrow |p|_{i+1} > 0$, and τ_β is recorded when $|p|_i > 0 \rightarrow |p|_{i+1} = 0$. These intervals are segments of time during which the place owns tokens.
- For a transition t, τ_α is recorded when $\mathcal{G}(I)_i = 0 \rightarrow \mathcal{G}(I)_{i+1} = 1$, and τ_β is recorded when $\mathcal{G}(I)_i = 1 \rightarrow \mathcal{G}(I)_{i+1} = 0$. These intervals are segments of time during which the transition is enabled (and thus executing the firing function).
- For a token k, given that $k \in p_i$ at cycle i, τ_α and τ_β are recorded when $p_i \neq p_{i+1}$. Tokens can be owned by nil. These intervals are segments of time describing how long the token has been owned by any given place.

Such historical data can be useful for making certain turn-taking decisions. For example, one can decide whether to act based on the amount of time spent acting previously; this is used to simulate the user parameter of initiative, described in Section 5.4. Although decisions based on timing history are non-Markovian and can be difficult to analyze closed-form using currently available mathematical techniques, simulation can be used to analyze systems of arbitrary complexity.

3.3 Application to Multimodal Reciprocal Interactions

Figure 2 depicts a simplified example of how multimodal state is represented in our system. The tokens are numbered and labeled with their values. In our system visualization, we also show filled and partially filled transition rectangles, which communicate firing delay expectations. In the example, the robot is about three quarters of the way through executing a "wave" animation, has just finished executing text-to-speech of the phrase "Hello," and is about to start looking at the human partner.

Figure 3 depicts the base set of dependencies between all such modules in our system, which are implemented as Petri net subgraphs. The edges in the diagram indicate that modules interface by connecting Petri net primitives. When the graph is extended to support new domains, additional dependencies between modules may be introduced. The context model needs to be instantiated on a per-domain basis, as it selects semantic actions to be handled by the behavioral layer; these actions are timed by the turn-taking module. Our approach to this division within the interactional layer is further described in (Thomaz & Chao, 2011). We emphasize that the loosely denoted layers do not imply an strict order of execution, as a Petri net represents a distributed event system.

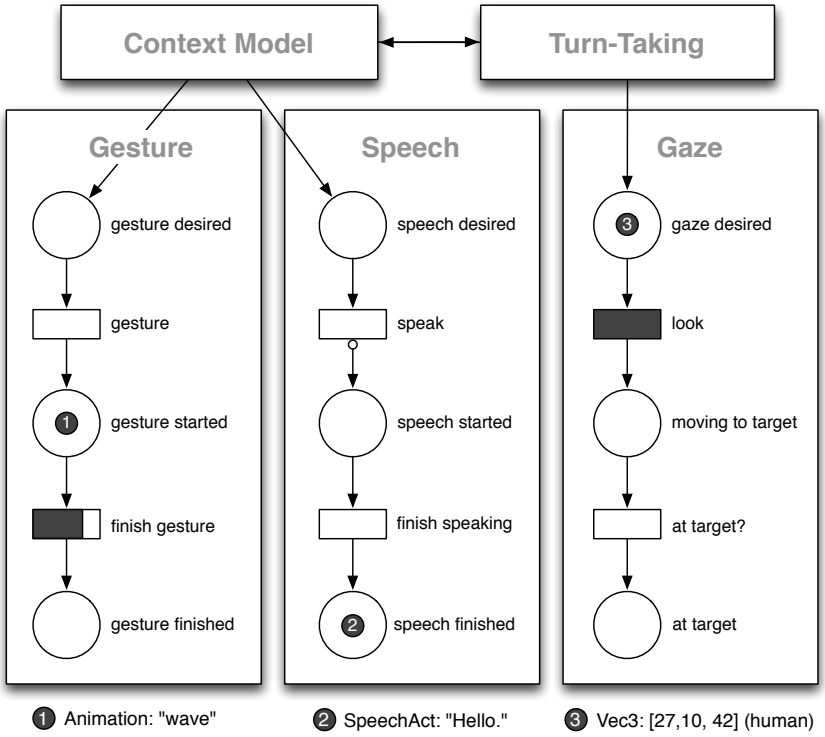


Figure 2. A simplified example of how multimodal state is represented in the Petri net.

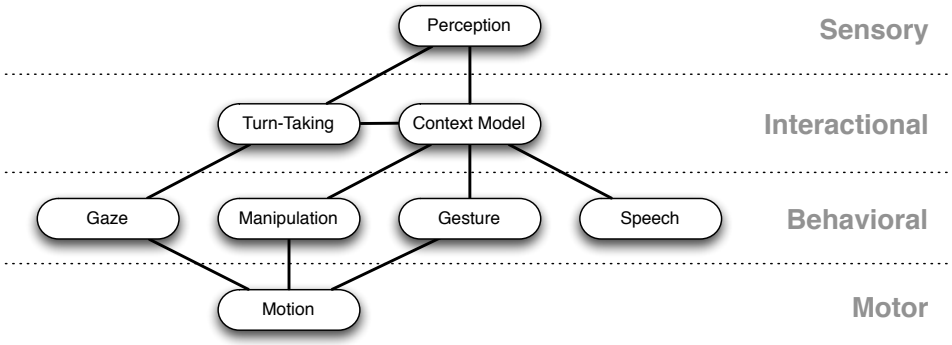


Figure 3. Dependencies of modules within our system.

4. System Extension: Action Interruptions

In this section, we describe what we hypothesize is one fundamental skill for robots to interact fluently with humans: the ability to interrupt temporally extended actions at arbitrary times. We motivate this design principle in Section 4.1 and describe its implementation details for our system in Section 4.2 utilizing the formalism introduced in Section 3. Sections 5 and 6 then focus on evaluating this addition.

4.1 Motivation

One contributor to fluency in humans is the equalized management of shared resources during cooperation — a distinct characteristic of human social activity (Warneken, Lohse, Melis, & Tomasello, 2011). When two humans share a bowl of popcorn or hold doors open for each other, they engage in seamless turn-taking of shared spaces. The conversational "floor" is another important shared resource. When humans converse to exchange information, they yield the floor when appropriate (Schegloff, 2000; Duncan, 1974). In the presence of shared resources, this continuous give-and-take process is necessary to balance control between two partners and thus maximize their contributions to the joint activity.

We believe that robots can achieve higher interaction fluency by using an action execution scheme that dynamically yields shared resources and control to humans in order to maintain efficient and reciprocal turn-taking. Humans should be able to exert fine-grained control over robots with the same kinds of subtle mechanisms used to influence other humans. On an extreme level, humans do have ultimate authority over robots through the emergency-stop button, but such coarse levels of control are not helpful for accomplishing cooperative tasks.

Here, we investigate how a robot can effectively yield control of two specific resources — the speaking floor and shared space — in the form of speech and manipulation action interruptions. We hypothesize that managing shared resources in this way leads to improved interaction balance and thus better task performance.

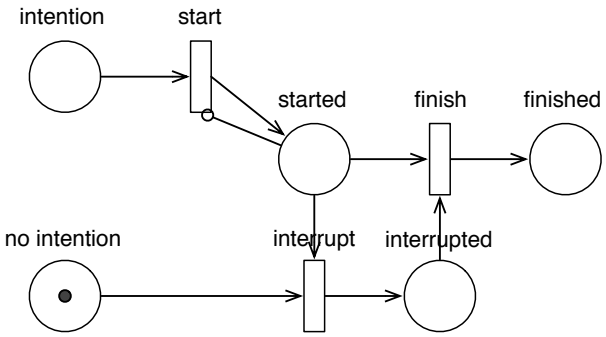
4.2 Implementation

4.2.1 Action Atomicity in Reciprocal Interaction Traditionally in HRI scenarios, basic-level actions such as gestures, gaze directions, and speech commands are triggered in response to stimuli and then executed to completion. For example, detecting a human reactively produces an action — an emblematic gesture of "wave," or a text-to-speech greeting such as "Hello, how are you?" One limitation of always completing communicative acts, as we discovered in previous work (Chao et al., 2011), is that it potentially adds superfluous bottlenecks to the interaction. It is also misleading regarding the internal state of the robot. Continuing to speak when the message is already across implies that the speaker believes the listener has not received the message.

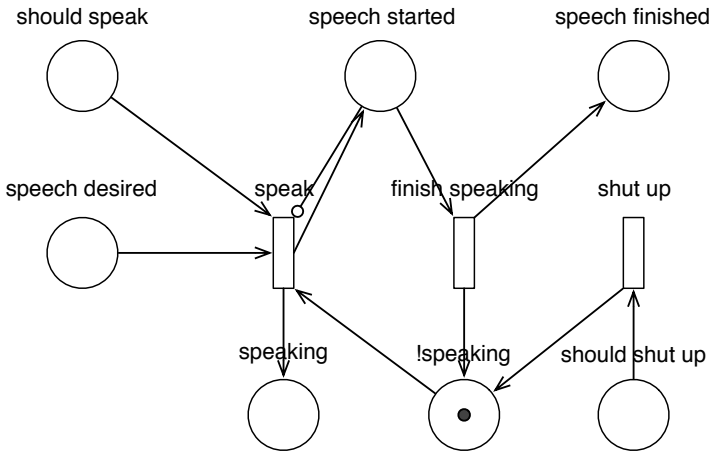
Other systems have endeavored to address this topic. The work in (Raux & Eskenazi, 2009) and (Nakano et al., 2005) are examples of dialogue systems in which speech interruptions in particular are supported. Interruption has also been addressed more indirectly through an approach of behavior switching (Kanda, Ishiguro, Imai, & Ono, 2004). We believe that interruption should be explicitly handled as its own intention ("stop this" rather than "react to something higher-priority"). In our implementation, we also support interruptions through multiple modalities of behavior. And our model is expressed formally in order to control complex system states precisely and facilitate further analysis. Often in ad hoc behavior-switching architectures, the modules interact in unpredictable ways that lead to "emergent" behavior that is difficult to understand or recreate.

The general idea is that an action should be divided into key stages. An atomic action should not be to "wave," but instead to "start waving," "keep waving," and "stop waving" at arbitrary points in time depending on the task context. You might continue waving at a friend across the street until he

catches your eye, after which you can stop immediately, because the necessary information has most likely been transmitted. This shift in action atomicity seems necessary to achieve fluency. Existing interactions with robots do not break down constantly, as some amount of fluency emerges naturally from simple reactive behaviors (Kose-Bagci et al., 2008), but they require a human to adapt his timing to the robot and back off when needed. Because the robot's behavior is not reciprocal, the robot tends to dominate control of interaction timing. This may be acceptable in certain situations, but should occur purposefully within a larger process of joint intentionality, not as an accidental side effect of the robot's action formulation.



(a) An example template for a fluent action.



(b) A speech process reflecting the structure of the template.

Figure 4. Visualization of behavioral actions, which are subgraphs of the Petri net behavior system.

4.2.2 Template for Fluent Actions Figure 4(a) shows a generic template for a fluent action in the form of a Petri net subgraph. A separate process decides whether the agent should have the intention of performing the action; if so, it deposits a token in $p_{intention}$ and destroys the one in $p_{no-intention}$. This token triggers t_{start}, which deposits a token in $p_{started}$. After the action has started, if an external process deposits a token in $p_{no-intention}$, this in combination with $p_{started}$ triggers $t_{interrupt}$, which deposits a token in $p_{interrupted}$. From there, t_{finish} can trigger and deposit

a token in $p_{finished}$. If the action is not interrupted, the action terminates normally through t_{finish} without ever encountering the interrupt loop. External processes can attach transitions to $p_{finished}$ to read the token values, essentially subscribing to the message for the event finishing.

The action atomicity paradigm previously described is clearly manifested in this action template through the three separate transitions of t_{start}, $t_{interrupt}$, and t_{finish}.

Figure 4(b) shows an example behavior in the system, the speech process. The subgraph is similar in structure to the basic action template. One difference is the additional contextual parameter of $p_{speech-desired}$ describing the particular speech act to execute, which combines with the intention of $p_{should-speak}$ in order to trigger t_{speak}. Another is the presence of the additional variables of $p_{speaking}$ and $p_{!speaking}$, which contrasts with the notion of a speech act that has started or finished (because a speech act may contain pauses in speech between strings of utterances).

A goal of this line of research is to develop an inferential turn-taking model within our architecture, which requires defining an interface between context models and generic interaction behavior in terms of information flow. For example, a turn-taking model may decide whether the robot should or should not speak in the speech process. If the robot has information to pass, it can produce that utterance; otherwise, it can opt for a backchannel.

5. Evaluation Methodology

We demonstrate use of a TPN control scheme for turn-taking and the value of action interruptions. We use a particular domain, a collaboration to solve the Towers of Hanoi, as described in Section 5.1. Our evaluation is performed within this domain through two means. One is a traditional user study, the protocol for which is detailed in Section 5.3. The other is a simulation experiment made possible by the TPN, described in Section 5.4, which we believe is an interesting contribution of the TPN representation for HRI research.

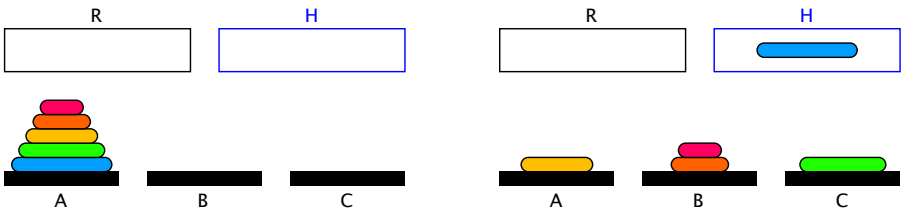
5.1 Domain: Collaborative Towers of Hanoi

To explore human-robot collaboration while staying in the realm of our robot's cognitive, perceptual, and physical capabilities, we chose to start with the classical artificial intelligence problem of the Towers of Hanoi. We intended for this abstract toy problem to be a metaphor for a collaborative workplace scenario in which a robot and a human need to cooperate in order to accomplish a physical goal, perhaps in a repetitive fashion, but while sharing certain resources such as tools and space.

The problem of the Towers of Hanoi is described by a set of three pegs and an ordered set of N pieces, usually disks of increasing size. The goal is to move the entire set of pieces from one peg to another by moving only one piece at a time. A piece cannot sit on any piece of lower ordinality than itself. For achievable manipulation with the robot, we instead used a set of equally sized cups that differed in color and relied on a color sequence to represent the ordering.

We modified the problem slightly to form a dyadic embodied collaboration. Each agent in the dyad is permitted to pick up a single piece at a time. However, pieces do not teleport to pegs instantaneously, as pick and place actions require time to execute in the real world. Thus, the state space is a vector of length N in which each value represents the owner of the piece: either a peg $\in \{A, B, C\}$ or an agent $\in \{H, R\}$. In this formulation, the human and the robot essentially serve as extra pegs, each with a capacity of one piece.

The state space is a directed graph in which nodes are the states described above. Nodes are connected with edges representing actions to pick up a piece or place an owned piece on a specific peg, or verbal requests for the human to perform either of these actions. Figure 6 shows the first four levels of reachability in the state space. A plan is determined using Dijkstra's shortest path algorithm on the state space graph. The robot replans the solution as needed whenever state changes are detected.



(a) Starting configuration. (b) Legal state in which the human is holding a piece.

Figure 5. Example state representations in collaborative Towers of Hanoi. A–C represent pegs, and H(uman) and R(obot) represent agents.

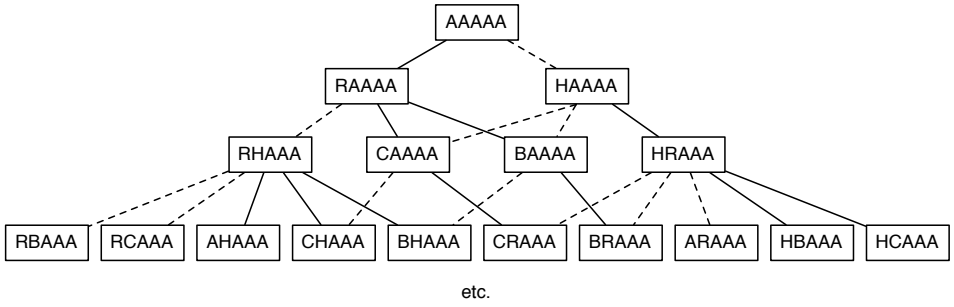


Figure 6. The first four levels of the collaborative Towers of Hanoi reachability graph for $N = 5$. Solid lines indicate manipulation actions, and dashed lines represent requests for human actions.

Towers of Hanoi is used in cognitive psychology tasks and can already be difficult for humans with $N = 4$ (Kovotsky, Hayes, & Simon, 1985). For $N = 5$ in our modified collaborative Towers of Hanoi, the solution is difficult for a human to see intuitively towards the beginning. About halfway through, the solution becomes much clearer, and at this point most humans are able to see the action sequence required to reach the goal.

5.2 Timed Petri Net Implementation of Domain

Figure 7 shows a system visualization of the implemented TPN for the Towers of Hanoi. A token in $p_{experimental}$ demarcates whether the robot should run $t_{backoff}$ or not, which activates the behavior for backing the robot's arm away from the shared workspace. This runs only when there is also a token in $p_{conflict}$ deposited by $t_{detectIntent}$, which is the perceptual process monitoring whether the human's hand has entered the shared workspace.

The transitions t_{act} and t_{move} differ in that t_{move} describes physical movement, and t_{act} describes the cognitive actions of processing the task state and selecting a task action to progress towards the goal. The task action can be either a manipulation action (controlled by t_{move}) or a verbal request to the human. The place $p_{bottleneck}$ gets filled when the task state reaches a point when the robot is bottlenecking on the human performing a task action, at which point $t_{handleBottleneck}$ initiates the control chain for a verbal request by interfacing with the speech process. When a human's action removes the bottleneck, t_{thank} is run to thank the user using speech.

The places p_{moving} and p_{still} for robot motion and $p_{speaking}$ and $p_{!speaking}$ for robot speech are meta-indicators for robot state. Incoming transitions that fill these places can be thought of as

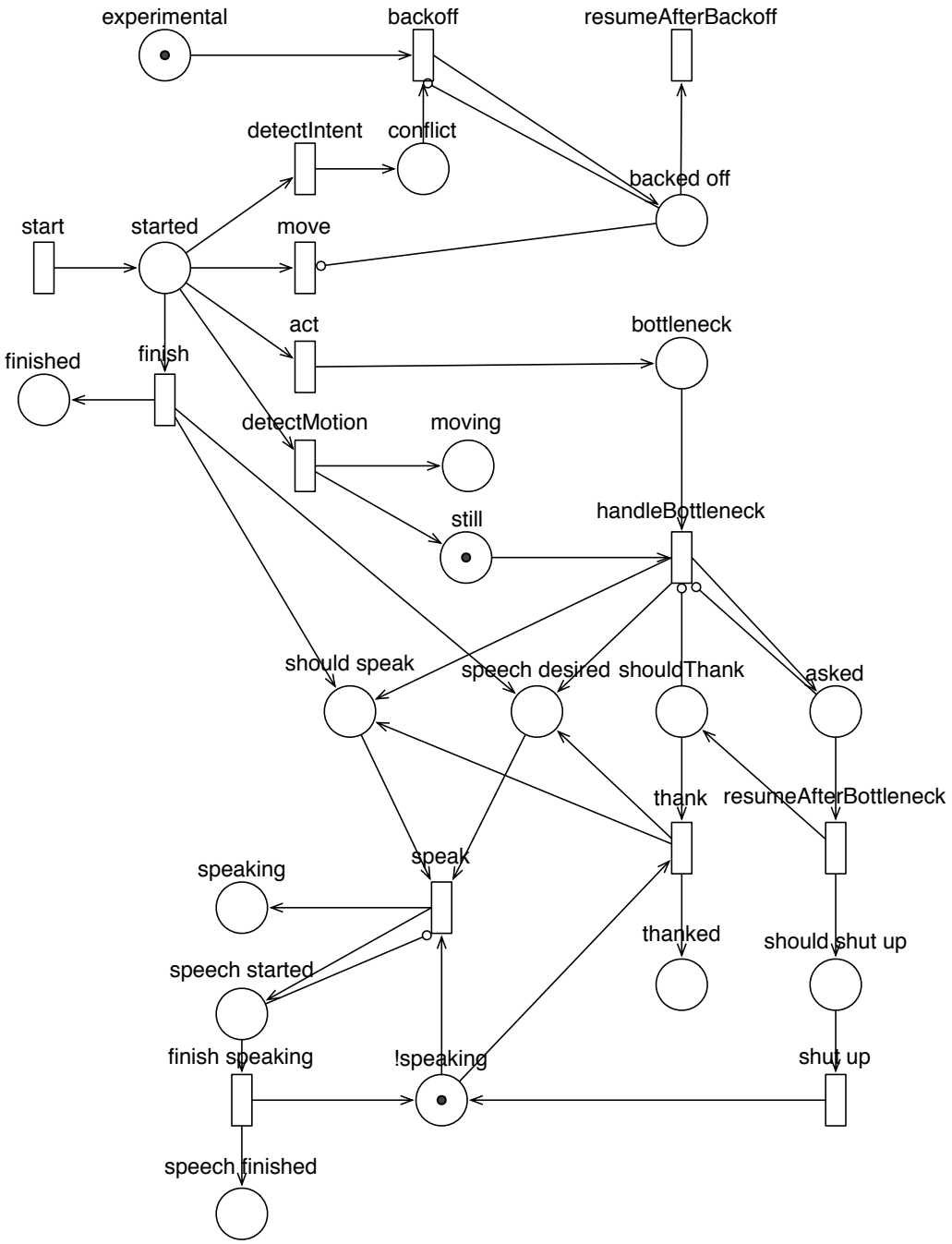


Figure 7. The system visualization of the timed Petri net used to control the robot in the Towers of Hanoi domain.

listeners. That is, $t_{detectMotion}$ does not control the robot to change the state of the external world; it simply internally monitors the robot's joints to determine if the robot is currently moving or not. So $t_{handleBottleneck}$ synchronizes on p_{still} and $p_{bottleneck}$ before running, meaning that it waits for a certain duration after the robot is no longer moving and after a bottleneck has existed for some amount of time before generating a verbal request to the human.

5.3 User Study

We designed and conducted an experiment to evaluate the effects of action interruptions within the system. The experiment was a between-groups study in which 16 participants collaborated with our humanoid robot Simon to solve the Towers of Hanoi problem.

5.3.1 Robot Platform The robot used for this study was our lab's upper-torso humanoid robot Simon (Figure 8(a)). Simon has two 7-DOF arms, 4-DOF hands, and a socially expressive head and neck. The arms and hands have compliant series-elastic actuators for safe operation alongside a human. For the study, the robot manipulated objects using only its right arm. Arm reach configurations were programmed on the robot to look natural using a kinesthetic learning from demonstration method, and object picking and placing actions were accomplished using torque-controlled power grasps and releases. Simon has two speakers near the base of the torso for communication through text-to-speech. It also has two ears containing LED arrays that are used to convey attention to colored objects.

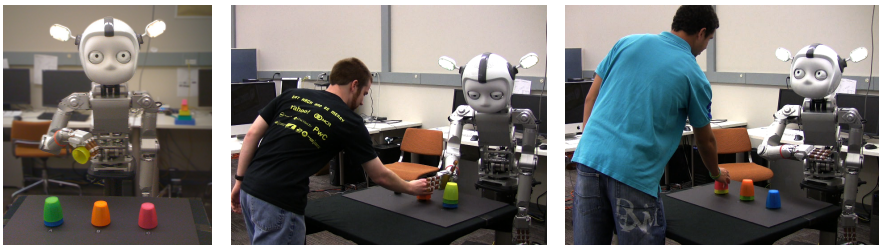
5.3.2 Environment Setup In the experimental setup, the participant stands across from the robot. The two are separated by a 34-inch square table that is covered by a black tablecloth (Figure 8). The table has three pegs rigidly affixed to a black foam board, and the pegs are equidistant from the positions of the human and the robot. The black foam board is designated as the shared workspace. Five differently colored, equally sized plastic cups are used as the Towers of Hanoi pieces. A stack of differently colored, differently sized blocks stands on a table behind Simon and serves as a mnemonic to help the participant remember the color sequence and the goal configuration.

Perception of the Hanoi pieces is done using an overhead camera pointing at the table. Because only the top piece is perceivable from this position, inferences about state need to be made based on these observations. Important perceptual events are color changes at peg locations, which could indicate any agent (the robot or the human) either removing a cup or placing a cup. Legal states consistent with the visual data and with the robot's intentions are preferred.

A structured light depth sensor, the Microsoft Kinect, is mounted on a tripod positioned adjacent to the robot and facing the human. The Kinect is registered to the robot's world coordinate frame, and the Kinect SDK is used to detect and track the participant's head and hand positions. This allows the robot to gaze at the participant's head pose, as well as to detect where participants' hands are in relation to the pegs. Specifically, the robot detects when the participants' hands enter and leave the shared workspace, as indicated by the region of the black foam board. When a human hand is in the shared workspace, the nearest peg is assumed to be the target of the human's next manipulation action.

5.3.3 Experiment Conditions The study is a between-groups design containing two conditions. The robot operated autonomously in both conditions:

- *Interruption condition* – In this condition, the robot interrupts its actions in response to the human. When performing reaching actions towards a particular peg, if the human's hand is in the shared workspace and approaches the direction of that peg (as detected by the Kinect skeleton tracker) then the robot interrupts its reach and switches its eye gaze to the human



(a) Workspace from participant perspective.

(b) Baseline condition.

(c) Interruption condition.

Figure 8. Simon backs off from the shared space in the interruption condition but not in the baseline condition.

(Figure 8(c)). When performing speaking actions to request an action from the human, it interrupts its speech if the human performs an action before the robot finishes speaking.

- *Baseline condition* – In this condition, the robot always runs reaching and speaking actions to completion before proceeding (Figure 8(b)).

5.3.4 Protocol Participants in both conditions were given identical instructions. After the Towers of Hanoi task was explained, participants were told that they were going to solve it collaboratively with Simon. They were instructed to use only one arm and to move only one piece at a time. They were encouraged to try to solve and execute the task quickly and in a parallel fashion with the robot. They were told that the robot might ask them to do some actions, but they did not have to listen to him, since the robot's world state could be prone to perceptual errors. They were also told that if Simon made any manipulation errors (e.g. failed to release a cup properly over a peg), that they should simply pick up the dropped cup and restore the state to a legal configuration.

Execution of the collaborative task lasted roughly five minutes per participant, and video was taken of the participants with their consent. Timestamped data of task start, completion, and state changes were logged throughout the interactions (and later confirmed or corrected through video analysis). After interacting with Simon, participants completed a survey containing the following questions:

1. On a scale from 1-100, how much did you contribute towards mentally solving the puzzle? (50 means both contributed equally)
2. On a scale from 1-100, how much did you contribute towards physically solving the puzzle? (50 means both contributed equally)
3. Please rate the following statements about the task. (1 = strongly disagree, 7 = strongly agree)
   (a) The task would be difficult for me to complete alone.
   (b) The task would be difficult for Simon to complete alone.
   (c) The task was difficult for us to complete together.
   (d) My performance was important for completing the task.
   (e) Simon's performance was important for completing the task.
4. Please rate the following statements about the interaction with Simon. (1 = strongly disagree, 7 = strongly agree)

(a) Simon was responsive to my actions.

(b) Simon was team-oriented.

(c) I trusted Simon's decisions.

(d) I had influence on Simon's behavior.

(e) Simon had influence on my behavior.

(f) I had to spend time waiting for Simon.

(g) Simon had to spend time waiting for me.

(h) We were efficient in completing the task.

(i) The interaction pace felt natural.

(j) There were awkward moments in the interaction.

5. (Open-ended) Please provide a critical review of Simon as a team member. Imagine that Simon is being evaluated at a workplace.

5.4 Simulation Experiment

With the same TPN system used to the control the robot in the experiment described in Section 5.3, we developed a simulation experiment to investigate the effects of user tendencies on the system dynamics. This experiment was conducted after a preliminary analysis of the results from the user study, as we believe it is important for simulations to be grounded in real user behavior. Qualitative observations of participants during the study pointed to certain factors being important to the task outcome. In particular, we observed that participants strategized differently about how to approach the problem. Certain participants made frequent, seemingly exploratory actions, sometimes undoing their most recent action or holding on to a piece at a peg while scratching their chins. Others were content to stand back and wait for the robot to tell them exactly what to do. In addition, participants tended to move faster than the robot, but most were worse than the robot at planning, especially at the beginning of the interaction.

Based on these observations, we developed a simulated user as a Petri net subgraph connected to the robot control graph. The simulated user subgraph includes the following components, as depicted in Figure 9:

- $P = \{p_{acting}, p_{idle}, p_{comply}\}$, where p_{acting} indicates a move has started, p_{idle} indicates that a move has finished, and p_{comply} indicates that a robot's request should be complied with. When p_{acting} has tokens, the human hand is detected as being near the peg of the selected move.

- $T = \{t_{start}, t_{finish}, t_{comply}\}$, where t_{start} selects and starts a move, t_{finish} ends a move, and t_{comply} decides whether to comply with a robot's request.

- $I = \{p_{acting} \rightarrow t_{finish}, p_{comply} \rightarrow t_{start}, p_{acting} \rightarrow t_{start}, p_{Speech} \rightarrow t_{comply}\}$, where p_{Speech} is a place in the robot graph indicating that robot speech has started.

- $O = \{t_{start} \rightarrow p_{acting}, t_{finish} \rightarrow p_{idle}, t_{comply} \rightarrow p_{comply}\}$.

We varied the simulated user's behavior along the following dimensions:

- *Speed* – the amount of time taken by the user per move (1–6 seconds). This controlled δ_f for t_{finish}, the time that tokens spent in p_{acting} before t_{finish} fired.

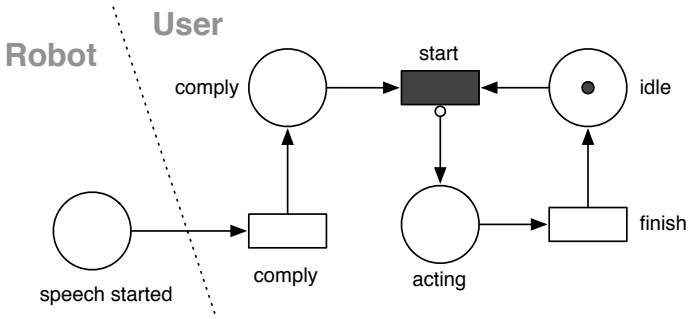


Figure 9. The simulated user behavior.

- *Initiative* – percentage of the time the user spends performing task actions in the shared space (0–50%). The transition t_{start} triggered when the following expression exceeded the initiative value, given by the intervals in p_{acting}:

$$\frac{\sum_i \tau_{\beta_i} - \tau_{\alpha i}}{\tau} \tag{1}$$

- *Compliance* – probability of the user complying with a robot's verbal request by performing the requested action (0–1), used in the control of t_{comply}.

- *Correctness* – probability of the user picking moves that are closer to the goal rather than randomly from the legal options (0.5–1), used in the control of t_{start}.

The experiment included 200 interaction runs, 100 each per experimental condition described in Section 5.3.3. Each run was produced by sampling parameter values uniformly at random from the specified ranges. The Petri net was run using a clock at 10x speed; that is, all actions were correspondingly sped up, including the robot motion, rate of text-to-speech, wait times, etc. The full experiment took approximately 6.4 hours to run (64 interaction hours). We terminated any given user run after 30 interaction minutes, even if it was not yet finished; this could occur as a result of poor strategies, timing, or deadlocking.

This experiment has two metrics of interest:

- *Execution duration* – the total time taken to complete the task.

- *Task balance* – the percentage of the final plan (action sequence) contributed by the human as opposed to the robot.

6. Results and Discussion

The results from the user study and simulation experiment are reported here. Our hypothesis was that action interruptions would increase interaction fluency by improving the balance of control between the robot and the human. That is, by relinquishing control to the human appropriately, the robot could allow the human to make better contributions to the task. We hypothesized that such a shift in balance would improve overall task performance, which should be observed as shorter task execution times. Additionally, we expected people to have a more positive perception of the robot and of the interaction in the interruption condition.

6.1 User Study Analysis

Results from the user study indicated that action interruptions resulted in reduced task execution time, perception of increased human contribution, and perception of fewer awkward moments in the interaction.

6.1.1 Task Efficiency The human-robot teams took significantly less time to complete the task in the interruption condition when participants had more control of the workspace ($M = 3.43$ minutes, $SD = 0.69$), as compared to baseline ($M = 4.56$ minutes, $SD = 1.55$), $t(7) = -1.9$, $p < .05$. Although the robot's planner attempted to minimize the number of moves, the goal was to optimize completion time, and humans were faster at executing actions but cognitively could not see many moves ahead.

There was no significant difference in plan length across the conditions, but the robot contributed significantly fewer moves in the interruption condition ($M = 8.75$, $SD = 1.71$) compared to the baseline ($M = 13.25$, $SD = 4.47$), $t(7) = -2.81$, $p = .01$. This led to a marginally significant differences in the ratio of human moves to robot moves (interruption: $M = 1.79$, $SD = 0.87$; baseline: $M = 1.19$, $SD = 0.31$, $t(7) = -1.79$, $p = .06$). These show the human's increased control when the robot used fluent actions.

There were also significantly fewer robot verbal requests in the interruption condition ($M = 3.5$, $SD = 1.58$) than in the baseline ($M = 7.25$, $SD = 2.54$), $t(7) = -4.02$, $p < .01$. Since there was no significant difference in human compliance to requests and compliance was high overall (94%), the robot contributed even more of the solution in the baseline through these requests.

6.1.2 Perception of Contribution Two survey questions concern the relative contribution of each team member to the success of the task. The relative contributions are parametrized along two dimensions: mental and physical. The mental contribution pertains to the algorithmic solution to the problem, and physical contribution pertains to the execution of it. For each dimension, we categorized people's numerical (1–100) responses about their own contributions as *equal to* ($= 50$), *more than* (> 50), or *less than* (< 50) that of the robot.

The distributions across conditions for relative physical contributions were the same; participants were well aware of their superior manipulation capabilities and movement speed compared to the robot. However, participants in the interruption condition were statistically more likely to state that their mental contribution was equal to or higher than the robot's compared to the baseline condition, $\chi^2(2, N = 8) = 6.17, p = .05$. This agrees with a result from our previous work, in which more submissive robot behavior results in better mental models for the human (Cakmak, Chao, & Thomaz, 2010).

In the baseline condition, the robot's increased tendency to monopolize the space above the pegs could have given the impression of always knowing what to do next. Although this was true to some extent, the robot's world state was still subject to perceptual errors, and the robot's plan did not take into account differing speeds of manipulation, so more human intervention would often have been beneficial. In addition, the yielding behavior may have communicated willingness to consider the human's strategy, making users more open to taking initiative. One participant in the baseline condition declared that the robot was *"a soloist"* and *"should be more of a team player."* Another observed, *"Simon was solving it so well... it did not feel like teamwork."* In the interruption condition, the robot's backing off from the shared space allowed the human to exert control over the plan being used to reach the goal. One participant said, *"I helped guide him to the solution to the puzzle quickly,"* asserting his belief of who was in charge.

6.1.3 Interaction Fluency One of the questions asked participants to rate their agreement or disagreement, on a 7-point Likert scale, with the following statement: *"There were awkward moments*

in the interaction." Although the notion of awkwardness may be difficult to quantify, we posed the question because we thought it would be intuitive for humans to answer. People in the interruption condition were less likely to agree that there were awkward moments in the interaction ($M = 3.75$, $SD = 1.71$) as compared with participants in the baseline condition ($M = 5.50$, $SD = 1.71$), $t(7) = 2.64$, $p = .03$.

We interpret this result as indicating a higher degree of interaction fluency in the interruption condition. Interruptible actions increased the impression of a reciprocal interaction, in which transparent intentions modulated behavior in both directions. A participant in the interruption condition commented: *"Simon... allowed me to make moves when I wanted while at the same time being decisive when he saw that I was pausing. He is an extremely good team member."*

6.2 Simulation Analysis

To analyze the simulation results, we ran ANOVA for the five factors described in Section 5.4 to characterize the impact of and interaction between factors for the observed variables of *execution duration* and *task balance*. The experimental condition was treated as a categorical variable (presence or absence of interruptions), and the others as continuous variables. These results are reported in Table 1. There were significant main effects for the manipulation of speed, initiative, and correctness for the simulated user. This implies that purely from the system standpoint, these human factors are observably important in determining task success and relative task contribution. Compliance with the robot's requests does not appear to have a significant effect. And although one might be inclined to assume that action interruptions should automatically improve task completion time by spending less time on unnecessary actions in general, the experimental condition (using interruptible actions) alone does not seem to have a significant effect, although there is a marginally significant interaction between the presence of interrupts and the initiative of the user ($p = .05$).

Table 1: ANOVA for simulation experiment results on the factors of condition, speed, initiative, compliance, and correctness. Execution duration describes the total time taken to complete the task, and task balance describes the percentage of the final plan (action sequence) contributed by the human as opposed to the robot.

Factor	**Execution Duration**	**Task Balance**
Condition	$F(1, 199) = 0.67, p = .41$	$F(1, 199) = 0.65, p = .42$
Speed	$F(1, 199) = 23.12, p < .0001$	$F(1, 199) = 25.05, p < .0001$
Inititative	$F(1, 199) = 17.85, p < .0001$	$F(1, 199) = 11.15, p < .01$
Compliance	$F(1, 199) = 0.09, p = .77$	$F(1, 199) = 0.1, p = .76$
Correctness	$F(1, 199) = 6.36, p = .01$	$F(1, 199) = 11.02, p < .01$
Cond × Speed	$F(1, 199) = 0.96, p = .33$	$F(1, 199) = 0.01, p = .91$
Cond × Init	$F(1, 199) = 3.61, p = .05$	$F(1, 199) = 0.16, p = .69$
Cond × Comp	$F(1, 199) = 0.11, p = .75$	$F(1, 199) = 0.32, p = .57$
Cond × Corr	$F(1, 199) = 0.06, p = .80$	$F(1, 199) = 0.48, p = .49$
Speed × Init	$F(1, 199) = 12.8, p < .001$	$F(1, 199) = 22.98, p < .0001$
Speed × Comp	$F(1, 199) = 0, p = .96$	$F(1, 199) = 1.21, p = .27$
Speed × Corr	$F(1, 199) = 10.7, p < .001$	$F(1, 199) = 8.71, p < .01$
Init × Comp	$F(1, 199) = 0, p = .98$	$F(1, 199) = 0.08, p = .78$
Init × Corr	$F(1, 199) = 35.68, p < .0001$	$F(1, 199) = 12.99, p < 0.001$
Comp × Corr	$F(1, 199) = 0.04, p = .85$	$F(1, 199) = 0.17, p = .68$

These simulation results only tell the story of the dynamics of the system. What they do not

indicate is the actual tendencies of real users. It still remains to be seen whether the uniformly distributed parameter space in the simulation experiment at all accurately represents a random selection of users in the real world. In addition, the essential question is whether the condition manipulation induces different behavior in users, resulting in differing distribution of such parameters across the groups.

To answer this, it is necessary to characterize the parameters of the study participants in same format as the simulation experiment. The amount of time participants spent performing actions in the shared space was annotated by two coders, including one of the authors. This was annotated as segments of time during which a given participant performed manipulation in the workspace. The intercoder agreement was 93.5%, describing the percentage of time that the annotation matched between the coders (between manipulating or not manipulating). The sum of times spent over these segments divided by the total task time yielded the initiative parameter; the average of this value was taken between the two coders. To determine the average time taken per action (speed), the number of actions per segment was also annotated with discussion between coders to resolve differences. Correctness was determined by whether the resulting game state of each human action was closer to the goal than the preceding state; the sum of closer actions was divided by the number of human actions. Compliance was determined by whether the next action taken by the human after a robot verbal request was the requested action, divided by the number of requests.

The resulting parameter values are shown in Table 2, and a comparison of the parameters tested in simulation and the parameters of the human participants is visualized in Figure 10. As can be seen from the figure, the parameters of participants from the user study were well within the span of values tested in simulation. As shown in Table 2, speed, compliance, and correctness did not differ across the groups, but initiative differed significantly. On average, subjects in the interruption condition spent 50% more of the total task time performing actions as compared to the baseline condition. We thus conclude that the presence of action interruptions leads users to take more initiative in the task, leading to the observed increase in task efficiency and improved balance of control.

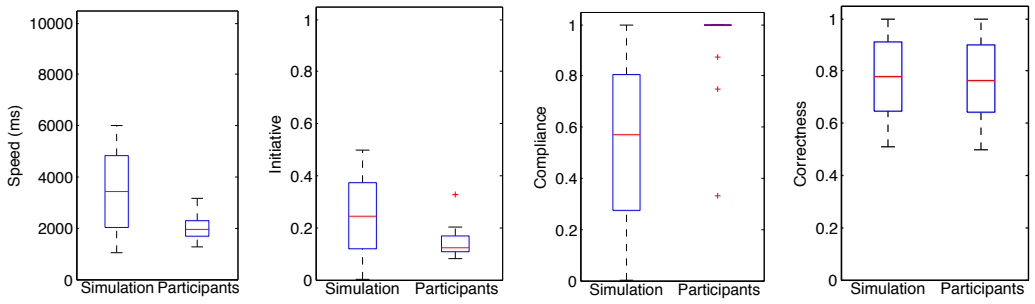
Table 2: Shown are the speed, initiative, compliance, and correctness for the participants in the user study across the two conditions. The parameter values were determined from logs and video coding

Parameter	**Baseline**	**Interruption**	**Significance**
Speed (secs)	1.6 (0.2)	2.0 (0.4)	$t(7) = -1.32, p = .11$
Initiative (%)	10.8 (2.2)	16.2 (6.9)	$t(7) = -1.96, p < .05$*
Compliance (%)	96.9 (8.3)	90.1 (21.8)	$t(7) = -0.79, p = .23$
Correctness (%)	71.1 (16.1)	79.5 (15.6)	$t(7) = -0.50, p = .32$

6.3 Generalizability of Results

In this domain, we have demonstrated an example of how our system simulation can augment the analysis of a user study. An obvious caveat with all simulation work is that assumptions made in simulation may not generalize well to interactions with human participants in the real world. All of human behavior cannot be summarized in a handful of parameters, and the parametrizations can be oversimplifying. There is also always the question of whether it is worth channeling effort into developing more accurate user models when user studies are needed anyway.

However, we do believe that iterative analysis of user data and simulation data can provide more perspective into how certain results were attained. The ability to run large quantities of simulations in much less time than would be needed for user studies is a powerful tool for developing and un-



(a) Speed (b) Initiative (c) Compliance (d) Correctness

Figure 10. Comparison of user parameters between those sampled in the simulation experiment and those from human participants in the user study.

derstanding the robot system. It also allows for broader coverage of parameters that may occur less commonly in the recruited user populations, which can be useful for stress-testing or finding corner cases. Because human-robot interactions are so complex in state and timing, they tend to be difficult to pigeonhole into closed-form mathematical techniques. We think that system characterizations through simulation can offer a scalable way to understand fluency in the face of such complexity. In this work, we have shown the ease with which a TPN framework allows such simulation experiments.

Another issue concerns the generalizability of our specific domain to other collaboration scenarios. One could argue that the inability of the robot to optimize for time rather than plan length was a failure of foresight and programming. In addition, the robot was slower and less dexterous than the human, which may not generalize to tasks where the opposite is true; in those cases, could the human's increased control be detrimental? It certainly should not be assumed that our outcome of reduced execution time universally generalizes to future tasks. However, the simulation technique allows for the rapid analysis of any adjusted dynamics. For example, if it became possible for the robot to move faster than the human, then the experiment could be quickly rerun after modifying the controller. In addition, any preprogrammed optimization ability of the robot may not optimize the human's goals at a particular moment. Humans adapt easily to fluctuating goals, and allowing robots to be able to yield appropriately results in spontaneous flexibility of the dyadic system, providing users greater control over this adaptive optimization process.

7. Conclusion

Our research focuses on developing robot behavior that improves fluency in multimodal reciprocal interactions. Toward this end, we developed a system for the control and analysis of timing in turn-taking interactions based on a timed Petri net representation. We described the semantics of our system and used it to implement a human-robot collaboration scenario based on the Towers of Hanoi. We hypothesized that action interruptions would contribute to fluency in reciprocal interactions, so we developed a fluent action template that we used as the basis for developing speech and manipulation behaviors. To examine the role of interruptions, we used a novel evaluation mechanism combining two types of experiments. We ran a user study with 16 participants in which the robot was autonomously controlled by our system, and we ran a simulation experiment that simulated 200 such users parametrized by the factors of speed, initiative, compliance, and correctness. Our analysis of results from both experiments showed that our implemented action interruptions

increased task efficiency primarily through a mechanism of increasing the initiative of the human partner. This resulted in the perception of improved interaction balance, which subsequently led to a reduction in time needed to complete the task.

Acknowledgements

This work was supported by ONR Young Investigator Award #N000140810842. The authors would also like to thank Karl Jiang for his contributions to video coding the data.

References

Barrett, L. R. (2010). *An architecture for structured, concurrent, real-time action*. Unpublished doctoral dissertation, University of California, Berkeley.

Bohus, D., & Horvitz, E. (2010). Facilitating multiparty dialog with gaze, gesture, and speech. In *Proceedings of the 12th international conference on multimodal interfaces (ICMI-MLMI)* (pp. 5:1–5:8, http://dx.doi.org/10.1145/1891903.1891910).

Bratman, M. (1992). Shared cooperative activity. *The Philosophical Review*, *101*(2), 327–341, http://dx.doi.org/10.2307/2185537.

Burgoon, J., Stern, L., & Dillman, L. (1995). *Interpersonal adaptation: Dyadic interaction patterns*. Cambridge University Press.

Cakmak, M., Chao, C., & Thomaz, A. L. (2010). Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, *2*(2), 108–118, http://dx.doi.org/10.1109/TAMD.2010.2051030.

Cao, T., & Anderson, A. C. (1993). A fuzzy Petri net approach to reasoning about uncertainty in robotic systems. In *Proceedings of the ieee international conference on robotics and automation (ICRA)* (pp. 317–322, http://dx.doi.org/10.1109/ROBOT.1993.292001).

Cassell, J., & Thorisson, K. R. (1999). The power of a nod and a glance: envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence*, *13*, 519–538, http://dx.doi.org/10.1080/088395199117360.

Chao, C., Lee, J. H., Begum, M., & Thomaz, A. (2011). Simon plays Simon says: The timing of turn-taking in an imitation game. In *IEEE international symposium on robot and human interactive communication (RO-MAN)* (pp. 235–240, http://dx.doi.org/10.1109/ROMAN.2011.6005239).

Duncan, S. (1974). On the structure of speaker-auditor interaction during speaking turns. *Language in Society*, *3*(2), 161–180, http://dx.doi.org/10.1017/S0047404500004322.

Hoffman, G., & Breazeal, C. (2004). Collaboration in human-robot teams. In *Proceedings of the 1st AIAA intelligent systems conference.*

Holroyd, A., Rich, C., Sidner, C., & Ponsler, B. (2011). Generating connection events for human-robot collaboration. In *IEEE international symposium on robot and human interactive communication (RO-MAN)* (pp. 241–246, http://dx.doi.org/10.1109/ROMAN.2011.6005245).

Holroyd, A. G. (2011). *Generating engagement behaviors in human-robot interaction*. Unpublished master's thesis, Worcester Polytechnic Institute.

Kanda, T., Ishiguro, H., Imai, M., & Ono, T. (2004). Development and evaluation of interactive humanoid robots. In *Proceedings of the IEEE* (Vol. 92, pp. 1839–1850, http://dx.doi.org/10.1109/JPROC.2004.835359).

Kose-Bagci, H., Dautenhan, K., & Nehaniv, C. L. (2008). Emergent dynamics of turn-taking interaction in drumming games with a humanoid robot. In *Proceedings of the 17th IEEE international symposium on robot and human interactive communication* (pp. 346–353, http://dx.doi.org/10.1109/ROMAN.2008.4600690).

Kovotsky, K., Hayes, J., & Simon, H. (1985, April). Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology*, *17*(2), 248–294, http://dx.doi.org/10.1016/0010-0285(85)90009-X.

Kozima, H., Michalowski, M. P., & Nakagawa, C. (2009). Keepon: A playful robot for research, therapy, and entertainment. *International Journal of Social Robotics*, *1*(1), 3–18, http://dx.doi.org/10.1007/s12369-008-0009-8.

Lacerda, B., & Lima, P. (2011). Designing Petri net supervisors from LTL specifications. In *Proceedings of robotics: Science and systems (RSS).*

Murata, T. (1989). Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE* (Vol. 77, pp. 541–580, http://dx.doi.org/10.1109/5.24143).

Mutlu, B., Shiwa, T., Ishiguro, T. K. H., & Hagita, N. (2009). Footing in human-robot conversations: How robots might shape participant roles using gaze cues. In *Proceedings of the 2009 ACM/IEEE conference on human-robot interaction (HRI)* (pp. 61–68, http://dx.doi.org/10.1145/1514095.1514109).

Nakano, M., Hasegawa, Y., Nakadai, K., Nakamura, T., Takeuchi, J., Torii, T., et al. (2005). A two-layer model for behavior and dialogue planning in conversational service robots. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3329–3335, http://dx.doi.org/10.1109/IROS.2005.1545198).

Raux, A., & Eskenazi, M. (2009). A finite-state turn-taking model for spoken dialog systems. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics (NAACL)* (pp. 629–637, http://dx.doi.org/10.3115/1620754.1620846).

Rich, C., Ponsler, B., Holroyd, A., & Sidner, C. L. (2010). Recognizing engagement in human-robot interaction. In *Proceedings of the 2010 ACM/IEEE conference on human-robot interaction (HRI)* (pp. 375–382, http://dx.doi.org/10.1109/HRI.2010.5453163).

Sakita, K., Ogawara, K., Murakami, S., Kawamura, K., & Ikeuchi, K. (2004). Flexible cooperation between human and robot by interpreting human intention from gaze information. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (Vol. 1, pp. 846–851, http://dx.doi.org/10.1109/IROS.2004.1389458).

Schegloff, E. (2000). Overlapping talk and the organization of turn-taking for conversation. *Language in Society*, *29*(1), 1–63.

Shah, J., Wiken, J., Williams, B., & Breazeal, C. (2011). Improved human-robot team performance using Chaski, a human-inspired plan execution system. In *Proceedings of the 6th international conference on human-robot interaction (HRI)* (pp. 29–36, http://dx.doi.org/10.1145/1957656.1957668).

Thomaz, A., & Chao, C. (2011). Turn-taking based on information flow for fluent human-robot interaction. *AI Magazine Special Issue on Dialogue With Robots*, *32*(4).

Wang, J. (1998). *Timed Petri nets: Theory and application.* Springer.

Warneken, F., Lohse, K., Melis, A., & Tomasello, M. (2011). Young children share the spoils after collaboration. *Psychological Science*, *22*, 267–73, http://dx.doi.org/10.1177/0956797610395392.

Weinberg, G., & Blosser, B. (2009). A leader-follower turn-taking model incorporating beat detection in musical human-robot interaction. In *Proceedings of the 4th ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 227–228, http://doi.acm.org/10.1145/1514095.1514149).

Zhang, W. (1989). Representation of assembly and automatic robot planning by Petri net. *IEEE Transactions on Systems, Man and Cybernetics*, *19*(2), 418–422, http://dx.doi.org/10.1109/21.31045.

Authors' names and contact information:

Crystal Chao, School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA. Email: cchao@gatech.edu.

Andrea L. Thomaz, School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA. Email: athomaz@cc.gatech.edu.