Action parsing and goal inference using self as simulator

Jesse Gray, Cynthia Breazeal, Matt Berlin, Andrew Brooks, Jeff Lieberman
MIT Media Laboratory
77 Massachusetts Avenue
Cambridge, MA, USA
[jg, cynthiab, mattb, zoz, xercyn]@media.mit.edu

***Abstract*— An important element of cooperative behavior is the ability to understand a teammate's actions, as well as to infer their goals and other mental states that are responsible for generating those actions. Simulation Theory argues in favor of an embodied approach whereby humans reuse parts of their own cognitive structure for not only generating one's own behavior, but also for simulating the mental states responsible for generating that behavior in others. This paper presents our simulation-theoretic approach and demonstrates its performance in a collaborative task scenario. The robot offers its human teammate assistance by either inferring the human's beliefs states to anticipate their informational needs, or inferring the human's goal states to physically help the human achieve those goals.**

I. Introduction

People are readily able to infer the mental states of humans and other animate entities in order to predict and understand their behavior. This competence is referred to by many different names in the scientific literature: theory of mind, mindreading, mind perception, social commonsense, folk psychology, and social understanding, among others. It is a foundational skill for many forms of human social intelligence including collaboration, deception, empathy, communication, and social forms of learning.

As robots enter the human environment and interact with novice users on a daily basis, we believe that robots must also exhibit social intelligence in order to communicate, cooperate, and learn from people. For instance, in a collaborative task scenario, a robot teammate could use belief and goal inferences to anticipate the human's needs (informational or instrumental) and to plan actions to provide the human with timely and relevant assistance. Towards this goal, we argue that robots will need to understand people *as people* — namely as social entities whose behavior is generated by underlying mental states such as intentions, beliefs, desires, feelings, attention, etc.

II. Methodology

Our approach to endowing machines with a similar capability is inspired by leading psychological theories and recent neuroscientific evidence for how human brains might accomplish this and the role of imitation as a critical precursor. This is in contrast to alternate theoretical viewpoints proposed from the study of autism (believed to be a deficit of theory of mind). For instance, Scassellati[17] implemented a hybrid model of work by Leslie[13] and Baron-Cohen[1] where joint attention is viewed to be a critical (and missing) precursor to the theory of mind competence.

Specifically, *Simulation Theory* holds that certain parts of the brain have dual use; they are used to not only generate our own behavior and mental states, but also to predict and infer the same in others. To understand another person's mental process, we use our own similar brain structure to simulate the introceptive states of the other person[6]. For instance, Gallese[9] proposed that a class of neurons discovered in monkeys, labeled mirror neurons, are a possible neurological mechanism underlying both imitative abilities and Simulation Theory-type prediction of the behavior of others and their mental states. Further, Meltzoff and Decety[15] posit that imitation is the critical link in the story that connects the function of mirror neurons to the development of mindreading. In addition, Barsalou[2] presents additional evidence from various social embodiment phenomena that when observing an action, people activate some part of their own representation of that action as well as other cognitive states that relate to that action.

Inspired by these theories and findings, our simulation-theoretic approach and implementation enables a humanoid robot to monitor an adjacent human by simulating his or her behavior within the robot's own generative mechanisms on the motor, goal-directed behavior, and perceptual-belief levels. This grounds the robot's information about the user in the robot's own systems, allowing it to make inferences about the human's likely goals and beliefs in order to predict the person's needs in accomplishing those goals.

III. Related Work

In related work in building computational systems, mirror neurons have inspired dual production-perception representation of motor skills for robots for skill transfer via learning from observation (e.g., [7]). In contrast, our work builds upon similar ideas and extends them to the cognitive level for mind perception skills.

Sophisticated statistical pattern recognition techniques have been applied to the recognition of human actions, gestures, etc. However, our approach is an effort to push beyond recognition of surface behavior to infer the internal states of the human that generate that behavior. This is of particular importance for predicting future behavior and for offering timely and relevant assistance in a teamwork situation.

Significant work in collaborative dialog and discourse theory has addressed intention inference and plan recogni-

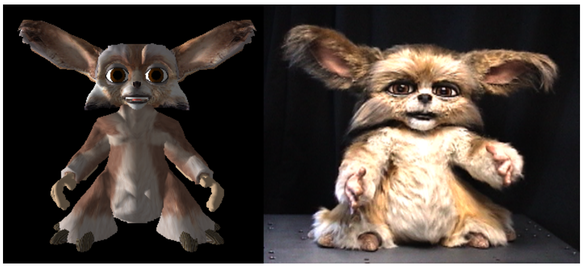


Fig. 1. The Leo robot and simulator

tion [11]. Whereas these systems rely on natural language processing to make these inferences, our work complements these past efforts to focus on what can be inferred from observing non-verbal behavior. Given our domain of human-robot interaction, understanding and making predictions based on people's non-verbal behavior is of particular importance to us. We have already demonstrated that humans apply their mindreading skills to infer the internal "mental" states of an expressive humanoid robot based on reading its non-verbal behavior (e.g., gaze direction, body orientation, facial expressions, hand trajectories, etc.), and how this improves the efficiency and robustness of human-robot teamwork [4]. Ideally, robots could read human non-verbal behavior to do the same — this work is a step in this direction.

Therefore, whereas there has been significant work in the areas of action recognition, perception-production of motor skills, plan recognition, intention inference, and belief inference — in our approach, the same methodology is used to handle each from an embodied and experiential perspective. This allows inferences made across these different systems to interact in interesting and useful ways - especially to support collaborative behavior. Furthermore, the robot's inferences are based on its grounded experience in its own interactions with the world and with people. This could contribute to making the robot's inferences more understandable and predictable to the humans who interact with it. Finally, reuse of existing systems to make such predictions has important and beneficial design implications for systems that have limited computational resources.

IV. Platform

The implementation presented in this paper runs on the Leonardo robot (Leo), a 63 degree of freedom humanoid robot (Figure 1). Leo sees the world through two environmentally mounted stereo-vision cameras. One stereo camera is mounted behind Leo's head for detecting humans within the robot's interpersonal space. The second stereo camera looks down from above, and detects objects in Leo's space as well as human hands pointing to these objects. Leo can use his eye cameras for fine corrections to look directly at objects or faces at a higher resolution. In order for Leonardo to perceive the pose and upper-torso motions of a human interacting with the robot, we use a Gypsy motion capture suit. This suit has potentiometers mounted near the joints of the human to determine their

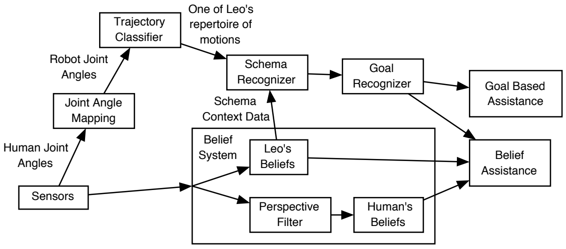


Fig. 2. System Architecture Overview - Inferring beliefs allows for informational kinds of assistance over only inferring actions and goals to provide instrumental assistance.

joint angles (represented as Euler angles). This suit is worn by the human interacting with Leo (see figure 9). In the future, we plan to replace the use of this suit with a vision solution to allow for more natural interactions.

V. Overview: Architecture for Simulation Theory

Our implementation computationally models simulation-theoretic mechanisms throughout several systems within the robot's overall cognitive architecture. See Fig. 2 for a system diagram. For instance, within the motor system, mirror-neuron inspired mechanisms are used to map and represent perceived body positions of another into the robot's own joint space to conduct action recognition. Leo reuses his belief-construction systems from the visual perspective of the human to predict the beliefs the human is likely to hold to be true given what he or she can visually observe. Finally, within the goal-directed behavior system, where schemas relate preconditions and actions with desired outcomes and are organized to represent hierarchical tasks, motor information is used along with perceptual and other contextual clues (i.e., task knowledge) to infer the human's goals and how he or she might be trying to achieve them (i.e., plan recognition).

The general methodology is summarized as follows:

- Map the human's actions onto the robot's motor representations. Tag these actions as coming from "other."
- Use dual-pathways from the motor to the cognitive systems to pass this information to those systems that evoke these movements. For instance, movements used to achieve task goals would pass up to the goal-directed behavior system.
- Consider the perceptual context from the human's visual perspective. Based on this, hypothesize what their likely beliefs are about the perceptual context. Tag these as coming from "other".
- Consider any other relevant contextual information, such as task knowledge.
- With this dual-pathway information, use the cognitive systems as a simulator running on these "other-derived" inputs to infer the likely goals or beliefs that would arise given these circumstances within the associated systems. (Note that multiple hypotheses could be generated and weighted probabilistically to indicate how likely they are).

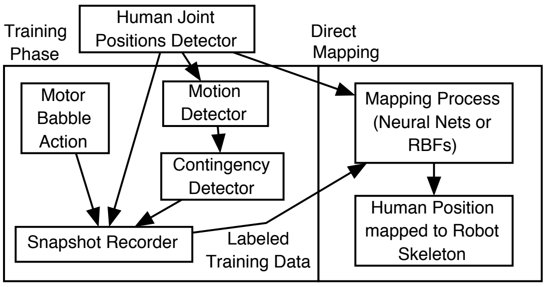


Fig. 3. Mapping perceived human joints onto the robot's skeleton to allow for comparison between joint configurations of robot and human

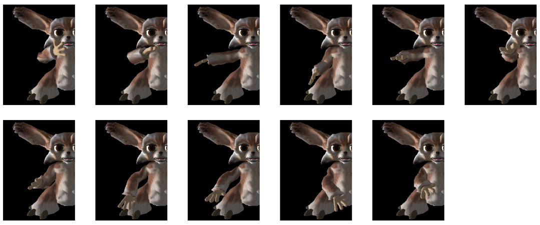


Fig. 4. Training poses for right arm.

- Use this information to predict the human's behavior, and to shape the robot's own responses.
- Incorrect inferences present an opportunity for learning. Ideally, even incorrect inferences should at some level seem plausible to the human. This will assist with efficient error recovery and reduce the chances the same mistake is made in the future.

VI. Action Recognition and Movement-Level Simulation

This section describes the process of using Leo's perceptions of the human's movements to determine which motion from the robot's repertoire the human might be performing. The technique described here allows the joint angles of the human to be mapped to the geometry of the robot even if they have different morphologies, as long as the human has a consistent sense of how the mapping should be and is willing to go through a quick, imitation-inspired process to learn this body mapping. Once the perceived data is in the joint space of the robot, the robot tries to match the movement of the human to one of its own movements (or a weighted combination of prototype movements). Representing the human's movements as one of the robot's own movements is more useful for further inference using the goal-directed behavior system than a collection of joint angles.

A. Body Mapping

Leonardo has the ability to physically imitate humans and to recognize and critique human movements. To imitate people, the robot needs to convert the 3D joint angle data it perceives about the human (wearing the motion capture suit) into its own 1D joint space (i.e., its actuators) in order to physically "map" the human's body onto its own. To analyze this movement, Leonardo must then convert this body mapping data into a representation that allows the robot to compare the observed human movement against prototype movements stored within the robot's movement repertoire.

To learn a mapping between the human's body and Leonardo's body, an example set consisting of pairs of tracked human body poses matched to robot poses is needed. To acquire this data, the robot engages the human in an intuitive "do as I do" imitation game, inspired by early facial imitation whereby human infants learn how to imitate the facial expressions of their caregivers [16]. In this scenario, the robot first takes the lead and moves through a series of poses as the human imitates. Because the human is imitating the robot, there is no need to manually label the tracking data – the robot is able to self-label them according to its own pose. Because the structure of the interaction is punctuated rather than continuous, the robot is also able to eliminate noisy examples as it goes along, requiring fewer total examples to learn the mapping.

We have found that it is often difficult for people to imitate the entire full-body pose of the robot at once. Rather, when imitating, people tend to focus on the aspect of the movement that they consider to be most relevant to the pose, such as a single limb. Furthermore, humans and Leonardo both exhibit a large degree of left-right symmetry. It does not make sense from either an entertainment or a practical standpoint to force the human to perform identical game actions for each side of the robot. We therefore divide the robot's learning module into a collection of multiple learning "zones". Zones which are known to exhibit symmetry can share appropriately transformed base mapping data in order to accelerate the learning process.

Once the robot finishes acquiring its training set, it applies a radial basis function mapping technique to train the inter-body mapping. We have found that the RBF does a good job in capturing interdependencies between the joints, and provides a very faithful mapping in the vicinity of the example poses. However, its performance decays as the data moves away from the training examples. As a result, the robot must use a carefully selected set of training poses to sufficiently cover the movement space. We found that this technique was also sensitive to errors during the training process, particularly in the case of closely neighboring example poses, which could sometimes cause significant warping of the map. A final list of poses that worked sufficiently well for us is shown in table 4.

When the robot relinquishes the lead to the human, the robot then tries to imitate the human's pose. The human can then take a reinforcement learning approach, verifying whether the robot has learned a good mapping or whether a further iteration of the game is required. One advantage of this entire process is that it takes place within an intuitive interactive game between the human and robot (instead of forcing the robot offline for a manual calibration procedure which breaks the illusion of life) and requires no special training or background knowledge on the part of the human teacher.

B. Action Parsing

For robots, the issue of how to segment a continuous stream of joint angles along discrete action boundaries is a challenge. The most commonly employed technique for the automatic segmentation of action streams is based on the observation that when humans engage in a complex action, they typically change direction and speed between each segment of that action, with an associated acceleration [12]. Matarić exploits this fact, by looking for changes in the overall motion of the joints involved in the robot [14]. To find episode boundaries, this technique works as follows. Generate the Mean Squared Velocity (MSV) of the joints, as shown in Eq. 1, where N represents the number of joints, and $q(i)$ is the position of joint i.

$$MSV(t) = \sum_{i=1}^{N} \left(\frac{dq(i)}{dt} \right)^2 . \tag{1}$$

For each time t, if $MSV(t-1) \leq c$ and $MSV(t) \geq c$, search through the remaining times until finding u such that $MSV(u) > kc$, for some c and k, chosen in advance. If a time u is found, then this value of t is an episode beginning. Switching inequalities yields the method to find episode endings. This technique basically looks for sufficiently low joint motion that reach sufficient highs, and vice versa.

Figure 5 shows the MSV during a complete button-push action, with marked episode beginnings and endings, for three trials. The episodes correspond to the reach, push down, release, and retract phases. The last two phases are analyzed as one episode, as the retraction from the button and the retraction back to the starting phase do not involve much slowdown.

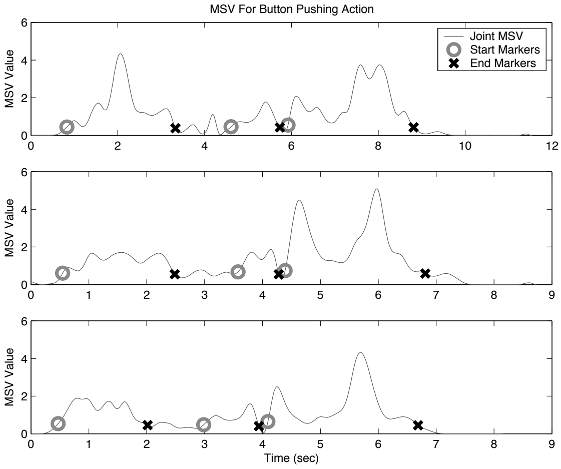


Fig. 5. MSV analysis of a button push, for three different trials. Episode beginnings and endings are marked as noted in the legend. As seen in the episode graphs, this separation corresponds with the reaching for the button, the press of the button, and the retraction from the button and return to the starting position

The values for c and k are typically manually tuned to create proper episodes. However, we have found in our experience that certain boundaries are often missed due to the lack of tactile feedback. For instance, sudden contact between the end effector and object usually signifies a new episode, but this is often missed, as motion does not necessarily cease when contact is first made. To remedy this shortcoming, we add to the MSV a tactile factor [scaled by $c_{tactile}$] to accommodate these changes:

$$MSV_{\text{improved, tactile}}(t) = \sum_{i=1}^{N} \left(\frac{dq(i)}{dt} \right)^2 + c_{tactile} \frac{dw(t)}{dt}, \tag{2}$$

where $w(t)$ represents the tactile sensor value at time t.

We also automatically generate the c and k parameters to compensate for overall high or low-motion actions. Recall that c represents the low-motion cutoff value, and k the ratio of high-motion to low-motion necessary for an episode. We choose c and k:

$$\begin{aligned} c &= < MSV > -\frac{1}{2}\sigma_{MSV}, && (3) \\ k &= \frac{< MSV > +\frac{1}{2}\sigma_{MSV}}{< MSV > -\frac{1}{2}\sigma_{MSV}}, && (4) \end{aligned}$$

such that $ck =< MSV > +\frac{1}{2}\sigma_{MSV}$, creating a symmetric distribution about the mean of the MSV. This automatically creates constants for the generation of several episode boundary markers.

Given the set of all possible markers for episode separation for each trial, we choose all possible episode boundaries from all trials, and test alignment between them to determine which markers are dominant through all trials. Once this marker subset space is searched, we mark true episode boundaries, and dynamically timewarp the trials to align them. A statistical correlation of the warped trials provides a measure of alignment between the two. For each trial, this ranking is added to a subset-specific list, to be combined with all of the iterations over all trial and subset combinations.

Whichever subset possesses the highest total is chosen as the true marker set. Figure [6] shows three trials involved in a button pressing action, the initial set of possible episode boundary markers for each, and then the final selection of chosen markers.

C. Matching Observed Actions to Motor Repertoire

The robot's motor system is represented as a graph of connected poses, called the *pose graph*. The pose graph is also used to perform the robot's repertoire of movements [8]. The nodes represent specific body (or facial) poses, and the arcs represent allowed transitions between them. Families of poses can be represented as a sub-graph of actions (e.g., kinds of walking, sitting, throwing, etc.) and links between sub-graphs represent allowable transitions between families of actions. In addition, weighted blends of either discrete poses or full trajectories can be generated to enlarge the repertoire of possible movements [8]. Finally,

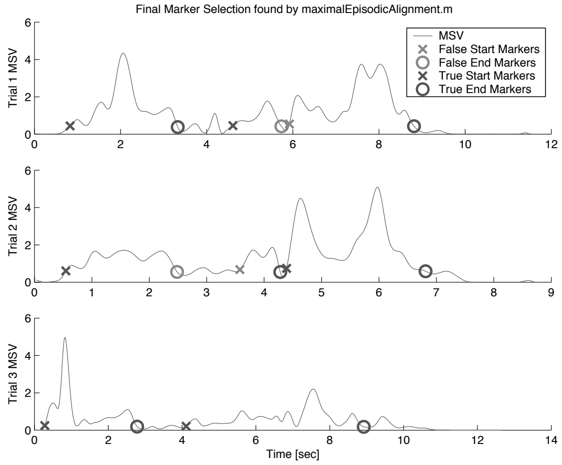


Fig. 6. A view of three trials involved in a button pressing action. The initial possible episode boundaries are marked in each example, as well as the final selection of markers, for use in combination for a canonical motion.

missing transition links can be substituted with inverse-kinematic solutions.

The robot can use its motor system as a resource to recognize observed actions of another. To do so, the segmented joint angle data is matched against trajectories through the robot's pose-graph to determine which of its own motions is most similar to the motion the human is performing. This involves searching over dynamically time warped joint trajectories to align the observed trajectory with those of the robot, and determine a minimal mean square error. We are currently exploring the use of Morphable Models to extend this concept [10], [5]. These constructs are designed to represent spatio-temporal trajectories in such a way that new variations can be synthesized through weighted linear combination of the known set of basis sequences, and observed sequences can similarly be matched through linear decomposition in terms of the basis set. For a complete mathematical treatment of Morphable Models, please see [10]; we provide a summary of our use of them here.

In order to compare and synthezize motion trajectories from the robot's own known actions, we must be able to simultaneusly compare the sequences in both space and time. For a given pair of (multi-dimensional) sequences, the correspondence problem is ill-posed: there is an infinite number of potential pairs of spatial and temporal displacement vectors that could "map" one sequence onto another. These can be separated into a linear class of spatio-temporal shifts by representing the tradeoff between the relative importance of spatial versus temporal consistency as a constant weighting over the entire sequence. As we are more concerned with the spatial content of actions rather than their precise timing, we have set this constant to be weighted heavily in favour of spatial correspondence — that is, when mapping one trajectory to another, large temporal shifts are permitted with little contribution to the overall error signal, whereas spatial displacements contribute significantly to this measure of inconsistency between examples.

The mapping problem thus becomes a constrained optimization problem — for a given spatio-temporal weighting, how can an optimally small combination of spatial and temporal displacements be generated while maintaining a strictly monotonic mapping of time from one sequence to the other? The Morphable Model representation makes use of a Dynamic Programming solution that is based on a sparse sampling of the features in time. In order to achieve real-time performance, we use a sampling scheme that produces 10 "keyframe" features per action sequence. Our matching algorithm therefore proceeds as follows:

- A baseline motion sequence is determined by averaging over the set of prototypical actions in the robot's movement repertoire. Linearly synthesized sequences are then able to be represented as spatio-temporal displacements from this sequence.
- For each example action sequence, an optimal spatio-temporal displacement sequence is calculated relative to the baseline sequence. These displacements are the morphable models of the actions.
- When an unknown action is demonstrated, it is resampled as above and its morphable model computed.
- To classify the demonstration, a linear parameter estimation problem must be solved in order to determine a set of weights for the known basis actions that would best replicate the demonstrated action. The resulting parameters can be used to identify the closest known action, or to theorize that the demonstration action is a synthesis of particular known actions.

To solve the system of equations that produce a set of action prototype weights, conventional linear least squares can be used. However, as the size of the basis set becomes large, linear least squares exhibits poor generalization properties. Giese and Poggio[10] showed that a better method to use in this case is one in which the capacity of the approximating linear set is minimized, and proposed the Modified Structural Risk Minimization algorithm to enable the use of *a priori* information as a constraint. In this technique a positive weight matrix is used to punish potential linear combinations of incompatible basis actions.

This approach suits our application, because the robot often has contextual knowledge of the types of actions that are likely to be appropriate in a given situation. For example, the action of pressing a button looks extremely similar to that of patting a stuffed animal, and the linear decomposition of an example reaching action could appear to be an indiscernible mixture of the two if all basis actions were allowed to be weighted equally. But by using the object referent (a button versus a stuffed toy) to select an appropriate weight matrix (that could be automatically generated from past experience), the robot is able to select a much more confident match.

VII. Goal Inference and Task Level Simulation

Our representation for goal-achieving behavior is intended to not only allow the robot to perform a set of actions to achieve a desired result, but to also introspect over them to determine the person's goal (based on what the robot's goal would be if it were performing the same action in the human's situation). Ideally the same information is needed to detect actions and make inferences as to perform them, so when actions are designed (and in the future, learned), only accomplishing the forward task need be taken into account and the detection and simulation will come for free. We follow this design principle except for one concession related to determining which object a human is acting upon (or other parameters relevant to the action). This mechanism requires a special detector, a *Parameter Generation Module*, as described below.

Within the deliberative system of the robot, the atomic-level representation of a goal-directed behavior is a schema that associates its necessary perceptual preconditions with a specific action (optionally performed on a particular object) to achieve an expected outcome (its goal). Schemas can be organized sequentially and/or hierarchically to create larger structures to represent tasks and execute them. When chaining sequential schemas, the goal of one schema becomes the precondition of the subsequent schema. Compound tasks are specified as a hierarchy of schemas, where the expected result of multiple schemas are the inputs (i.e., listed in the preconditions) of the subsequent schema. Therefore, to achieve some desired *task goal*, only the initial precondition for the first schema of the task model need be activated, and the system will traverse the hierarchy, completing preconditions and executing actions of each activated schema as necessary, until the final precondition is fulfilled (or it becomes known to the system that the final precondition cannot be fulfilled). See Figure 8 for an example.

When simulating the goal-directed behavior of others, the deliberative system begins by first attempting to determine which of its schemas might match the person's current contextual situation and action. Thus, one of the key inputs into this system is the action recognition result described in the previous section. Once the robot classifies an observed motion as matching one that if can perform, it triggers a search over its schemas to see if any would evoke that same motion from the movement repertoire. If multiple schemas involve the same motion (or if an observed motion matches multiple known motions with similar probability) the set of candidate schemas is narrowed by matching the current human's perceptual context against the necessary context for that schema.

Note, however, that the perceptual factors of the schema context must be from the perspective of the human (not of the robot). For example, if the robot is trying to determine whether the human is intending to turn a button ON, it adopts the visual perspective of the human to consider the button closest to his or her hand (rather than the one closest to its own hand). If the robot were to perform this button-activation schema on its own, the target button would be determined by a higher level decision making process.

However, the robot does not have access to this information from the human, so it must use a different mechanism to detect the target object of an observed schema. Each schema employs an associated *parameter generation module* which determines the parameters under which the human is performing the schema. For many schemas (such as button-activation) the only necessary parameter is the target, and a simple module that finds the object of a certain type that is closest to the human's hand is sufficient. For schemas that can be performed in multiple ways or on multiple objects, filling in these missing parameters allows the robot to more completely simulate the observed action. Failure of this module to find appropriate parameters indicates the context is not correct for performing the associated schema.

Once a schema has been selected as a suitable match, the robot infers that the human's immediate goal is the expected result of that schema. Since the robot has mechanisms to determine its own success or failure (by evaluating the goal-achievement condition for that schema), it can monitor the success of the human by running the same process with the relevant parameters from the human's action (as described above). In the case of a task goal, the robot can examine the preconditions of the schemas that comprise that task and track the human's progress over the course of the task. This is useful for the robot in order to anticipate the human's needs and offer relevant help accordingly.

VIII. Belief Inference and Visual Perspective Simulation

We believe that for successful cooperative interaction, modeling the human's beliefs about the task state is important given that these beliefs may diverge from those of the robot in dynamic and unpredictable circumstances. For example, a human may not be able to see important information due to a visual occlusion that is not blocking the robot's vantage point. To maintain coordinated activity, establishing and maintaining mutual beliefs (i.e., common ground) is important [11].

Leo keeps track of information about the world using his Belief System [3]. Data comes in from Leo's sensors, and passes through the robot's perception system. In the belief system, the robot sorts and merges perceptual feature data to determine what data corresponds to new information about present objects, and what data indicates the appearance of a new object. The object representation consists of a history of properties related to each perceived object.

In the spirit of simulation theory, we reuse this mechanism to model the beliefs of the human participant. For this, the robot keeps a second set of beliefs that are produced through the same mechanism — however, these beliefs are processed from the visual perspective of the human. This is different from simply flagging the robot's beliefs as human visible or not, because it reuses the entire architecture which has mechanisms for object permanence, history of properties, etc. By reusing the entire system you can get more complicated modeling: for example, this approach can keep track of the human's incorrect beliefs about objects that have changed state while out of the human's view. For instance, if the human sees a button that is ON before it is moved behind a occluding barrier and turns OFF - Leo's belief about the button is that it is OFF but the robot models the human's belief about the state of the button as still being ON. In this way, Leo has an initial set of mechanisms for modeling false beliefs.

This technique has the advantage of keeping the model of the human's beliefs in the same format as the robot's, allowing for direct comparison between the two. This is important for establishing and maintaining mutual beliefs in time-varying situations where beliefs of individuals can diverge over time. Currently we use the same processing to produce the human's beliefs as the robot's, except that we apply a visual perspective filter for the human's beliefs based on what they should and shouldn't be able to see - this allows the robot to keep a model of what the human believes about their view of the local world (i.e., the robot understands that what a person sees (or does not) influences what they can know).

Further changes could easily be made to the human's perceptual system - for example, if the robot believed that the human was colorblind, it could remove the color percept from its processing of the human perceptual information, and then none of the human beliefs would contain any color information. One could even imagine modeling what the robot thinks the human thinks the robot knows using this mechanism, by maintaining a third set of beliefs that go through the humans filter as well as one additional filter. We have not explicitly implemented this yet, but the mechanism could feasibly support this kind of second order inference.

IX. Demonstration

We have tested this framework in a collaborative scenario where Leonardo acts as a helpful partner to its human teammate. Leo and the teammate stand across from a shared workspace where three buttons are placed. Each button can be turned ON or OFF. In addition, there are occluding walls placed in the workspace such that certain parts can be viewed by one teammate but not by both. See Figure 7. Note in the figure that one of the buttons is occluded from view of Leo's teammate, but not from Leo. Furthermore, one of the buttons has a lock that must be disengaged before that button can be turned on.

Leo and the human are to collaborate to perform a shared task. The overall goal of this task is to activate all the buttons in the workspace (to turn them all to the ON position). Leonardo already has a task model for this activity. The role of the teammate in the joint activity is to turn the buttons ON (since the person as a broader reach), and Leo's role is to help the partner succeed. In what follows, we use this task to illustrate how Leonardo's mindreading skills allow it to predict the human teammate's needs in order to offer him or her timely and relevant assistance.

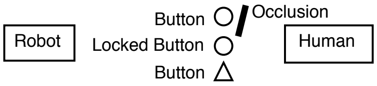


Fig. 7. Buttons task space. Turning on the locked button (center) requires first unlocking it (triangular button), then pressing it. One button is occluded from the human's view

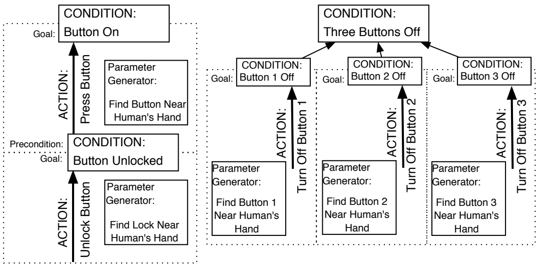
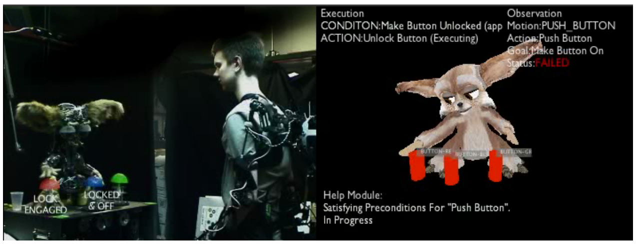


Fig. 8. Structure of two hierarchies of button related schemas. In forward operation, Leo (recursively) performs each connected schema until the task goal is satisfied. Leo detects a human doing one of his schemas if they are performing a similar trajectory and the parameter generator successfully finds appropriate action parameters (i.e., context is correct).

The robot's action recognition and goal inference capability is demonstrated when the human attempts to push one of the buttons but fails to activate it because it is locked. If the human performs a pushing motion (detected through the motion classification system) and their hand is near a button (detected through the parameter estimation system), the robot determines that they are trying to press that button (action recognition). Based on the structure of its own action in the same context and knowledge of the task, it infers that the person is trying to accomplish the goal of turning that button ON. Further, Leo evaluates the human's success by monitoring the state of the button, and tries to figure out how to best help the person succeed.

In the case of a locked button, Leo demonstrates that he can perform a helpful action by simulating the reason for the failure and noting that button is indeed locked. Instead of simply repeating the same action the human just performed (and also failing), Leo assists the human partner by acting to satisfy the precondition which the human overlooked, namely to disengage the lock. This allows the human to proceed with their goal to activate the troublesome button. Leonardo thereby demonstrates that the robot is goal-oriented and sensitive to the goals of the human partner, rather than simply action-oriented.

To demonstrate Leonardo's belief inference capabilities (to date this has been done using our graphical simulator) Leonardo assists the human overcome a visual occlusion that impedes successful completion of the shared task goal to turn all buttons ON. Recall that there is a barrier that blocks the human's view of one of the buttons. Leo keeps track of the human's beliefs through simulating their visual perspective of the scene to infer their likely beliefs. Leo concludes that because the human partner cannot see the occluded button, the human believes that there are



Execution
CONDITON:Make Button Unlocked (app
ACTION:Unlock Button (Executing)

Observation
Motion:PUSH_BUTTON
Action:Push Button
Goal:Make Button On
Status:FAILED

Help Module:
Satisfying Preconditions For "Push Button".
In Progress

Fig. 9. Human fails to activate a locked button (center). Leo notices the failure and assists the human in achieving his goal by disengaging the lock (image left).

only two buttons in this task instead of three. When the human begins the Turn-All-Buttons-ON task, Leo notices the actions using the above mechanism. The robot considers each required schema to determine if the human has enough information to succeed at the task. Noticing that the human cannot complete the compound action because of the obscured button, Leo performs an action to help them acquire the missing information. In this case, Leo points to the occluded button, thereby encouraging the human to look around the occlusion and to make visual contact with the hidden button.

X. Conclusion

We are pleased with the success of employing these simulation theoretic ideas within our architecture. One significant contribution of this work is to demonstrate how this methodology enables a robot to engage in various forms of ”mindreading” — beyond recognizing actions of a human partner to inferring their immediate goals, recognizing their task plans, and inferring their beliefs during a collaborative task. The second significant contribution of this work is to show how these abilities allow the robot to predictively offer relevant assistance in informational as well as instrumental terms to support the human partner's goals.

In the future, we would like to extend this architecture to allow for more varied inferences and interactions with humans. A limitation of the current system is that it matches the human's actions against exactly one of the robot's schemas. This works in the small examples we have created that do not have a large number of actions, but in the future we would like to take more advantage of the probabilistic information. Ideally this system would further incorporate the level of certainty of both the contextual information and the trajectory matching systems, as well as a short history of recent behavior of the human to try to match their behavior more accurately against the compound action structures. In the case of ambiguity, the system could wait and factor in the human's next moves to help resolve it. For cases where incorrect inferences are made by the robot (which we regard as inevitable given that humans also make incorrect inferences as well), we want to extend this implementation to allow the robot to learn from its past mistakes to avoid making the same errors in the future.

Furthermore, we would like to compare the inferences made by the robot in collaborative task scenarios to those inferences made by people in the same situation. This will be an important step in evaluating how human-compatible the robot's inferences are, and how relevant and timely its helpful actions are as compared to those offered by other people. To address this, we are preparing to conduct a series of human-human collaboration studies, and compare the results with subsequent human-robot collaboration studies. The implementation described in this paper is a critical step towards conducting these studies.

References

[1] S. Baron-Cohen. *Mindblindness*. MIT Press, 1995.

[2] L. W. Barsalou, P. M. Niedenthal, A. Barbey, and J. Ruppert. Social embodiment. *The Psychology of Learning and Motivation*, 43, 2003.

[3] Bruce Blumberg, Marc Downie, Yuri Ivanov, Matt Berlin, Michael Patrick Johnson, and Bill Tomlinson. Integrated learning for interactive synthetic characters. *ACM Transactions on Graphics*, 21(3: Proceedings of ACM SIGGRAPH 2002), 2002.

[4] C. Breazeal, C. Kidd, A. Lockerd Thomaz, G. Hoffman, and M. Berlin. Effects of non verbal communication on efficiency and robustness in human-robot teamwork. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), in Review*, 2005.

[5] Brooks, A.G., Berlin, M., Gray, J., and Breazeal, C. Untethered robotic play for repetitive physical tasks. In *ACM International Conference on Advances in Computer Entertainment*, 2005. (Submitted, under review).

[6] M. Davies and T. Stone. Introduction. In Martin Davies and Tony Stone, editors, *Folk Psychology: The Theory of Mind Debate*. Blackwell, Cambridge, 1995.

[7] J. Demiris and G. Hayes. Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in animals and artifacts*, pages 321–361. MIT Press, Cambridge, MA, 2002.

[8] M. Downie. Behavior, animation, and music: The music and movemnt of synthetic characters. Master's thesis, MIT, 2000.

[9] V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, 1998.

[10] Giese, M.A. and Poggio, T. Morphable models for the analysis and synthesis of complex motion pattern. *International Journal of Computer Vision*, 38(1):59–73, 2000.

[11] B. J. Grosz and C. L. Sidner. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in communication*, chapter 20, pages 417–444. MIT Press, Cambridge, MA, 1990.

[12] O. C. Jenkins and M. J. Matarić. Deriving action and behavior primitives from human motion data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2551–2556, 2002.

[13] A. M. Leslie. Tomm, toby, and agency: Core architecture and domain specificity. In L. A. Hirschfeld and S. A. Gelman, editors, *Mapping the Mind: Domain specificity in cognition and culture*. Cambridge University Press, Cambridge, 1994.

[14] Maja J. Matarić. Sensory-motor primitives as a basis for imitation: linking perception to action and biology to robotics. In *Imitation in animals and artifacts*, pages 391–422. MIT Press, Cambridge, MA, USA, 2002.

[15] A. N. Meltzoff and J. Decety. What imitation tells us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience. *Philosophical Transactions of the Royal Society: Biological Sciences*, 358:491–500, 2003.

[16] A. N. Meltzoff and M. K. Moore. Explaining facial imitation: a theoretical model. *Early Development and Parenting*, 6:179–192, 1997.

[17] B. Scassellati. Foundations for a theory of mind for a humanoid robot. *Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, PhD Thesis*, 2001.