Toward resilient human-robot interaction through situation projection for effective joint action

Adrian R. Pearce*, **Liz Sonenberg**** **and Paddy Nixon*****
*NICTA Victoria Research Laboratory,
Department of Computer Science and Software Engineering, The University of Melbourne
** Department of Information Systems, The University of Melbourne
***The University of Tasmania
Corresponding author: adrianrp@unimelb.edu.au

Abstract

In this paper we address the design of robots that can be successful partners to humans in joint activity. The paper outlines an approach to achieving adjustable autonomy during execution– and hence to achieve resilient multi-actor joint action – based on both temporal and epistemic situation projection. The approach is based on non-deterministic planning techniques based on the situations calculus.

Introduction

This work tackles the research challenges of *dynamic autonomy* and *information fusion in remote interaction* as described in the recent comprehensive survey of human-robot *interaction* research (Goodrich and Schultz 2007), and as exemplified in the growing interest in human-robot *co-operation* research, including with physical robots in complex operational settings, e.g. (Carbone et al. 2008; Crandall, Goodrich, and others 2005; Woods et al. 2004).

The work concerns multi-actor scenarios involving *mixed initiative* activity, sometimes termed *adjustable autonomy*, i.e. scenarios where the level of system autonomy may be adjusted on the run, by either the human or the agent, e.g. (de Greef and Arciszewski 2008; Johnson et al. 2010; Neef et al. 2009; Zieba and others 2011).

Ideal level of autonomy? Focusing on interactions where the appropriateness and effectiveness of system autonomy may vary as a situation unfolds, is an acknowledgement that the 'ideal' level of autonomy is not a static point along the spectrum from full teleoperation (i.e. complete human control of the agent) to full autonomy (i.e. no influence on the agent by the human operator), but is a point that will move, in both directions, along the spectrum as context and resources change (Crandall, Goodrich, and others 2005; Parasuraman, Sheridan, and Wickens 2000).

Resilient multi-actor interaction—the gap: For actors to make appropriate choices regarding information gathering, sharing, and action alternatives, they must be able to reason about present and future–intended actions required to meet their (shared) goal. We investigate the *hypothesis* that, for agents, a key to the establishment of effective interdependent activity is to combine situation projection into the future *(temporal projection)* with projection to the situations of other agents *(epistemic projection)* in an integrated reasoning framework. Our work lies in the intersection of research from the Cognitive Robotics community e.g. (Carbone et al. 2008; De Giacomo, Lespérance, and Levesque 2000; Pirri 2011; Levesque et al. 1997) with that of several groups, including prior work of the authors, from the Multi-Agent Systems community, e.g. (Johnson et al. 2009; Kelly and Pearce 2010; Klein et al. 2005; So and Sonenberg 2009; Ye et al. 2009).

The *specific aims* of this work draws on the authors' recent work e.g. (Kelly and Pearce 2010; So and Sonenberg 2009; Ye et al. 2009), to develop novel situation representations and planning algorithms that enable the computation of multi-actor perspectives including future environmental changes and awareness of other actors, and implement these in a new programming language, RoboGolog. Exercising and evaluation will employ a series of progressively more complex, simulated, human-agent challenge problems, e.g. *Blocks Worlds for Teams* (Johnson et al. 2009) and *Human in The Loop Simulation* (Kawamura and others 2006).

Illustrative Human-Robot Example: We utilise the well-known *blocks world* to illustrate. Imagine a set of coloured blocks sitting on a table. The goal is to build one or more vertical stacks of blocks in the correct order. The catch is that only one block may be moved at a time: it may either be placed on the table or placed atop another block. Only specific sequences of actions lead to the desired goal states. The computational complexity of this problem is well understood (Gupta and Nau 1992). In order to study joint activity of heterogeneous teams in a controlled manner, the basic blocks world problem has been extended to a human-robot setting (Johnson et al. 2009), the *Blocks Worlds for Teams* task, which caters for both multi-agent and heterogeneous (human-agent) teams.

Blocks Worlds for Teams: Colour blocks (or numbered boxes) are hidden in a series of rooms and agents can only observe blocks that are in the same room as they are. This forces the need for explicit coordination strategies and communication. In one room, the storeroom, there is a table. The team goal is to stack the blocks on the table to match the specified sequence as fast as possible. The diffi-

culty of this problem can be varied according to the initial conditions—there may be existing stacks of blocks on the table in the wrong desired sequence and the distribution of blocks among rooms can be initially changed.

To retrieve a block, an agent must maneuver into the various rooms to find the block of interest, then pick it up and maneuver into the storeroom and place it on the appropriate stack, in the correct order. When present, human team members use a computer interface. Agents communicate by sending messages to one another, restricted to a set of domain specific commands, to bound the interpretation problem for the artificial robot agents. This problem allows for a very diverse set of coordination techniques by the players during the course of achieving the goal (see (Johnson et al. 2009) for details).

Background

We follow Bradshaw's *Coactive Design* approach (Johnson et al. 2010), founded on the proposition that when people are working with agents, the (joint) processes of understanding, problem solving and task execution are "necessarily incremental, subject to negotiation, and forever tentative." From this perspective, the challenge is not to make agents more autonomous, but to make them more capable of being interdependent (Johnson et al. 2010), where *interdependence* is characterised by the requirement that, "what one party does depends on what another party does (and usually vice-versa), often over a sustained cycle of actions" (Klein et al. 2005). From this perspective, the design of agents (i.e. their internal architecture and their operational mechanisms) needs to be guided by an understanding of the interdependence in the joint activities they may undertake.

To achieve interdependence, agents must be able to recognise and manipulate the pertinent knowledge, beliefs, and assumptions that the involved actors share (Klein et al. 2005).

Multi-actor situation projection

The classical situation projection problem involves predicting a future situation based on the effects of performing a sequence of actions. Solutions to the projection problem have been achieved using the situation calculus, a widely adopted formalism for modelling the effects of actions through time. While sound and complete results have long been been known for this problem (Reiter 1992), with tractable solutions more recently demonstrated in domains such as *RoboCup* (Finzi and Orlandini 2005), current approaches do not adequately capture coaction or the effects of *multiple* actors. Projecting forward requires reasoning, not only about the possible effects of sequences of actions in a dynamic environment, by exploiting models provided at system design time about action effects *(temporal projection)*, but also about the possible actions (potentially either helpful or interfering) of the other actors in the scenario, by exploiting models provided at design time about their cognitive mechanisms – e.g. their belief structures and goal directed behaviours *(epistemic projection).*

Of importance here is a novel multi-agent version of the situation calculus developed by Pearce that has the capacity to express actions and their effects for multiple actors, and has been shown to enable projection in multi-agent case (Kelly and Pearce 2010). By itself this approach is not sufficient to solve human-agent projection since individual actors only know part of the complete situation, yet communication is limited by the cognitive constraints of human actors, necessitating local processing in the context of incomplete information. Under these circumstances the problem cannot be reduced to the single-agent case through knowledge sharing but must respect the distributed, epistemic nature of the problem. The interdependence of actors' actions requires actor-to-actor correspondence between relevant, reciprocal and mutual situations (Johnson et al. 2009; Keogh, Sonenberg, and Smith 2009) and in turn requires multi-faceted and flexible representations of situations (So and Sonenberg 2009; Thomson, Terzis, and Nixon 2006; Ye et al. 2009).

The situation calculus

The *situation calculus* is a logical language specifically designed for representing and reasoning about dynamically changing worlds. It has had a profound influence on the practice of achieving effective automated planning in robotics and has been successfully deployed in the *RoboCup* soccer tournaments (Ferrein, Fritz, and Lakemerer 2005). A possible world history is represented by a term called a *situation*. The constant S_0 is used to denote the initial situation where no actions have yet been performed. Sequences of actions are built using the function symbol do, such that $do(a, s)$ denotes the successor situation resulting from performing action a in situation s, e.g. the situation $do(pickUp(agt, block, S_0)$ results from performing action $pickUp(agt, block)$ in initial situation S_0 The representation of a dynamic world in the situation calculus consists of, *basic action theories* (Pirri and Reiter 1999) comprising of first-order logic statements capturing the following: axioms describing the *initial configuration* of the world; *actions* that can be performed in the world; *fluents* describing the state of the world, e.g. $Holding(agt, block, s)$, $Clear(block, s)$, $On(block1, block2)$; *precondition axioms* stating when actions can possibly be legally performed, e.g. $Poss(pickUp(agt, block), s) \equiv Clear(block, s)$, and *successor state axioms* describing how fluents change between situations, e.g.

$$Holding(agt, block, do(a, s)) \equiv a = pickUp(agt, block) \vee Holding(agt, block, s) \wedge a \neq putDown(agt, block)$$

Situation projection

The MINDIGOLOG (Kelly and Pearce 2006) language has the capacity to solve the classic (temporal) situation projection problem of predicting a future situation based on the effects of performing a sequence of actions. The language is a variant of the Golog family of languages (Levesque et al. 1997). It incorporates partial orders and has concurrency control (incorporating the abilities of (De Giacomo, Lespérance, and Levesque 2000)) and the ability to plan continuously using an efficient online search operator (based on (Sardina et al. 2004)). Domain-specific actions are connected by standard programming constructs, and that may contain nondeterministic operators. The primary advantage of this approach is controlled nondeterminism, allowing some program parts to be fully specified while others may involve arbitrary amounts of nondeterminism facilitating *flexible task specification* and *plan failure detection*. The approach utilises effective automated theorem proving techniques based on logical regression (see (Pirri and Reiter 1999) for details).

Flexible task specification. MIndiGolog combines powerful nondeterminism control, such as controlling the order of actions, who does what etc. The approach is based on a sophisticated logic of action that captures concurrent actions, continuous actions and an explicit representation of time. Some of the automated planning operators of the MIndiGolog language include: a, execute action a in the world $\phi?$, proceed only if condition ϕ is true; $\delta_1;\delta_2$, execute program $\delta_1$followed by δ_2; $\delta_1|\delta_2$, execute either δ_1 or δ_2; $\pi(x)\delta(x)$, nondeterministically select arguments for δ; $\delta*$, execute δ zero or more times; $\mathbf{if}\,\phi\,\mathbf{then}\,\delta_1\,\mathbf{else}\,\delta_2$, execute δ_1 if ϕ holds, δ_2 otherwise; $\mathbf{while}\,\phi\,\mathbf{do}\,\delta$, execute δ while ϕ holds; $\delta_1||\delta_2$, concurrently execute $\delta_1$and δ_2; $\delta_1 \gg \delta_2$, concurrently execute δ_1 prioritised over δ_2; $\Sigma\delta$, find and perform legal exec. of δ; and $\mathbf{proc}P(\overrightarrow{x})\delta(\overrightarrow{x})\mathbf{end}$, procedure definition (see (Kelly and Pearce 2006) for details). One powerful feature of the language is the ability to nondeterministically select an agent for a particular role. Consider the following procedure for an agent to collect a block to stack on the table in the storeroom:

$$\begin{array}{l}
\quad\quad \mathbf{proc}\, stackBlock(agt, colour) \\
\quad\quad\quad\quad \pi(room)(moveTo(agt, room))*; \\
\pi(block, Colour(block, colour)?; pickUp(agt, block)); \\
\quad\quad\quad\quad \pi(room)(moveTo(agt, room))*; \\
moveTo(agt, storeroom); putOnStack(agt, block)\, \mathbf{end}
\end{array}$$

There are potentially many ways to execute this program, depending on which actions are possible in the world and the theory of action to project forward to a *legal execution*. The agents are free to determine an appropriate strategy that satisfies fluents, based on constraints, such as the proximity of agents to rooms with known boxes. Consider the following program that concurrently collects and stacks a blue block onto a red block (omitting check of stack order for clarity):

$$\pi(agt1, stackBlock(agt1, blue)) >> \pi(agt2, stackBlock(agt2, red))$$

Crucially, the individual agents assigned to picking up blocks are not specified or allocated to each room a priori and the prioritised concurrency operator allows the order in which each individual agent performs each role is determined, non-deterministically, at run time, here with a preference for the red block to be collected first based on the desired order.

Plan failure detection. Consider the following exploration program that searches for victims and reports them to human (tele)operators. The robot can detect failure of its plan while in fully autonomous mode, in the event that it is idle for more than 60 seconds, and informs the human operator to regain control:

$$\begin{array}{l}
\quad \mathbf{proc}\, investigate(robot, building), \\
\pi(room)(moveTo(robot, room))*; \\
\quad\quad \pi(victim, ObserveVictim(victim, room)?; \\
inform(robot, human, Victim(room))); \mathbf{end} \\
\quad \mathbf{proc}\, explore(robot), \pi(t, time(t)?; \mathbf{while} \\
investigate(robot, building);\ \mathbf{do} \\
\quad\quad \pi(t_1, robotIdle(robot); time(t_1)?; \\
(t_1 - t < 60)?); inform(robot, human, regainControl); \mathbf{end}
\end{array}$$

Epistemic Projection

The existing MIndiGolog language and execution semantics has the capacity to reason about the knowledge of situations in a multi-agent environment—especially for more complex situations than above—based on knowledge of what is observable. In the approach, complex forms of knowledge can be established through joint observation (see (van Bentham, Pacuit, and Kooi 2006)) and this has lead to the demonstration of an effective approach to reasoning about knowledge based on the same logical regression techniques used for reasoning about future situations (see (Kelly 2008) for details). Consider the fragment of code for the coordination required between a robot and a human agent collecting blocks, with the aim to only communicating with one another when necessary (omitting observation details for clarity):

$$\begin{array}{l}
\quad\quad \mathbf{if}\, \mathbf{Knows}(robot, Red(nextStackBlock))\, \mathbf{then} \\
\quad\quad\quad \mathbf{if}\, \mathbf{notKnows}(robot, \\
Holding(human, Red(block)))\, \mathbf{then} \\
\quad\quad\quad\quad pickUp(robot, block); \\
inform(robot, human, Holding(agt, Red(block))
\end{array}$$

Ideally, for robot agents to effectively reason about human-robot interaction there will be agreement about the situations they are in at any instant—from both the robot agent's perspective and the human's perspective—and accordingly adjust the autonomy so that the human teleoperator is only interrupted when necessary. Clearly, this is only possible in practice in so far as the task is adequately described from the perspective of the robot agent, according to the axiomatisation of the task in the MIndiGolog language.

Situation models

The ability of the robot agent to project onto situations that the human is encountering now, and situations that the human will encounter in the future requires a canonical form of observation we term *joint observation*—in the sense that a single agent reasons about the observations of other agents (or humans) in a standard, canonical form, that facilitates reasoning about the joint observations of those agents (or humans). To this end, we will utilise the multi-faceted and flexible representations of situations, encoded through situation lattices as developed by Nixon (Thomson, Terzis, and Nixon 2006; Ye et al. 2009), together with a canonical representation of partial orders that has been demonstrated using prime event structures, by Kelly (Kelly 2008) (Pearce's PhD graduate), which has the consequence of capturing joint observations involving partial orders from different agents in canonical form, and also drawing on the analysis of computational models of situation awareness by So (So and Sonenberg 2009) (Sonenberg's PhD graduate). MIndiGolog's epistemic projection capabilities, together with the new situation models, are key to solving our challenge of achieving situation projection in both the temporal and epistemic dimensions.

Related Work

Similar to others, e.g. (Carbone et al. 2008; Finzi and Orlandini 2005; Pirri 2011), our research will be *model–based*, i.e. requiring executable models of the problem domain, the available actor actions, and an underlying interpreter. More specifically, to achieve the theoretical foundations of our aims we require:

- a programming language which allows the domain's principles to be represented, while the actors' procedural operations are expressed by high level partial programs which can be completed and adapted to the execution context by a program interpreter;
- models of the controllable activities, with causal and temporal relations explicitly represented; and models of the actors – incorporating what is to be specified about their cognitive mechanisms; and
- reasoning engines to monitor execution from the perspective of each actor, and to perform planning within the constraints of the actor-specific perspective.

In a mixed-initiative setting the aim of a model-based system is twofold: on the one hand the operator activities are explicitly modelled and interact with the control system; on the other hand, the model-based monitoring activity exports a view of the system readable by the operator (which may be agent or human), hence the operator can further influence the agent status in a suitable interface (although, as noted earlier, detailed interface issues to accommodate a human operator are outside the scope of this project, consideration of this future focus will influence our design).

At each time instant agents progress their understanding of the situation in terms of possible actions they can perform to progress towards their goal, contingent on their awareness of the world state and the activities of the other actors. At some point the amount of non-determinism in the system will tip over - meaning the agents can no longer automatically (provably) come up with a plan to achieve their goal and must demand input from the operator actor. We observe that due to non-determinism within the environment, and due to the limitations on adequately axiomatising complex domains or the behaviour of human operators, after a sufficient amount of time, a fully autonomous system will require intervention.

Conclusion

In summary, crucial to the capacity of a system to exhibit adjustable autonomy during execution, and hence to achieve resilient multi-actor joint action, is the role of both temporal and epistemic situation projection supported by computational models and mechanisms for representing situations and their context. The focus of this paper has been an approach to the development and validation of novel and appropriate models and mechanisms.

References

Carbone, A.; Finzi, A.; Orlandini, A.; and Pirri, F. 2008. Model-based control architecture for attentive robots in rescue scenarios. *Autonomous Robots* 24:87–120.

Crandall, J.; Goodrich, M.; et al. 2005. Validating Human-Robot Interaction Schemes in Multitasking Environments. *IEEE Transactions on Systems, Man, and Cybernetics - Part A* 35(4):438–449.

De Giacomo, G.; Lespérance, Y.; and Levesque, H. J. 2000. ConGolog, a concurrent programming language based on the situation calculus. *AIJ* 121(1–2):109–169.

de Greef, T., and Arciszewski, H. 2008. Combining adaptive automation and adaptive teams in a naval command centre. In *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction*, ECCE '08, 23:1–23:4. New York, NY, USA: ACM.

Ferrein, A.; Fritz, C.; and Lakemerer, G. 2005. Using golog for deliberation and team coordination in robotic soccer. *Kunstliche Intelligenz* I:24–43.

Finzi, A., and Orlandini, A. 2005. Human-robot interaction through mixed-initiative planning for rescue and search rovers. In *AI IA 2005: Advances in Artificial Intelligence*, 483–494. Springer.

Goodrich, M. a., and Schultz, A. C. 2007. Human-Robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction* 1(3):203–275.

Gupta, N., and Nau, D. S. 1992. On the complexity of blocks-world planning. *Artificial Intelligence* 56(2-3):223–254.

Johnson, M.; Jonker, C.; van Riemsdijk, B.; Feltovich, P.; and Bradshaw, J. 2009. Joint activity testbed: Blocks world for teams (BW4T). In *Engineering Societies in the Agents World X*, 254–256. Springer.

Johnson, M.; Bradshaw, J. M.; Feltovich, P. J.; Jonker, C. M.; Riemsdijk, B. V.; and Sierhuis, M. 2010. Coactive Design Why Interdependence Must Shape Autonomy. In *Cognition*, In press. 20pp.

Kawamura, K., et al. 2006. Robotic body-mind integration: next grand challenge in robotics. In *RO-MAN 2004. 13th IEEE Int.l Workshop on Robot & Human Interactive Communication*, 23–28. IEEE.

Kelly, R. F., and Pearce, A. R. 2006. Towards high level programming for distributed problem solving. In Ceballos, S., ed., *IEEE/WIC/ACM Intern. Conf. on Intelligent Agent Technology (IAT-06)*, 490–497.

Kelly, R. F., and Pearce, A. R. 2010. Property persistence in the situation calculus. *Artificial Intelligence* 174:865–888.

Kelly, R. F. 2008. *Asynchronous Multi-Agent Reasoning in the Situation Calculus*. Phd, The University of Melbourne.

Keogh, K.; Sonenberg, L.; and Smith, W. 2009. Coordination in Adaptive Organisations: Extending Shared Plans with Knowledge Cultivation. In Vouros, G.; Artikis, A.; Stathis, K.; and Pitt, J., eds., *Organized Adaption in Multi-Agent Systems*, 90–107. Springer, Lecture Notes in Artificial Intelligence, vol. 5368.

Klein, G.; Feltovich, P. J.; Bradshaw, J. M.; and Woods, D. D. 2005. Common ground and coordination in joint activity. In Boff, W. B. R. . K. R., ed., *Organizational Simulation*, 139–184. John Wiley & Sons, Inc.

Levesque, H. J.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. B. 1997. GOLOG: A logic programming language for dynamic domains. *JLP* 31:59–84.

Neef, M.; van Maanen, P.; Petiet, P.; and Spoelstra, M. 2009. Adaptive Work-Centered and Human-Aware Support Agents for Augmented Cognition in Tactical Environments. In *Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience*, 68–77. Springer.

Parasuraman, R.; Sheridan, T. B.; and Wickens, C. D. 2000. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A* 30(3):286–97.

Pirri, F., and Reiter, R. 1999. Some contributions to the metatheory of the situation calculus. *J. ACM* 46(3):261–325.

Pirri, F. 2011. The well-designed logical robot: Learning and experience from observations to the situation calculus. *Artif. Intell.* 175(1):378–415.

Reiter, R. 1992. The projection problem in the situation calculus: a soundness and completeness result, with an application to database updates. In *Proc. Int'l Conf. on AI Planning Systems*, 198–203.

Sardina, S.; De Giacomo, G.; Lespérance, Y.; and Levesque, H. J. 2004. On the semantics of deliberation in IndiGolog – From theory to implementation. *Ann. Math. Artif. Intell.* 41(2–4):259–299.

So, R., and Sonenberg, L. 2009. The roles of active perception in intelligent agent systems. In *Multiagent Systems for Society*, 139–152. Springer, Lecture Notes in Artificial Intelligence, vol. 4078.

Thomson, G.; Terzis, S.; and Nixon, P. 2006. Situation determination with reusable situation specifications. In *IEEE Int. Conf. on Pervasive Computing and Communications (PerCom 2006)*, 620–623.

van Bentham, J.; Pacuit, E.; and Kooi, B. 2006. Logics of communication and change. *Information and Computation* 204(II):1620–1662.

Woods, D.; Tittle, J.; Feil, M.; and Roesler, a. 2004. Envisioning Human-Robot Coordination in Future Operations. *IEEE Transactions on Systems, Man and Cybernetics, Part C* 34(2):210–218.

Ye, J.; Coyle, L.; Dobson, S.; and Nixon, P. 2009. Using situation lattices in sensor analysis. In *IEEE Int. Conf. on Pervasive Computing and Communications (PerCom 2009)*, 1–11.

Zieba, S., et al. 2011. Using adjustable autonomy and human-machine cooperation to make a human-machine system resilient - Application to a ground robotic system. *Information Sciences* 181(3):379–397.