Hierarchical Reinforcement Learning for Communicating Agents

Michael Rovatsos
School of Informatics
University of Edinburgh
Edinburgh EH8 9LE, Scotland, UK
mrovatso@inf.ed.ac.uk

Felix Fischer Gerhard Weiss
Department of Informatics
Technical University of Munich
85748 Garching, Germany
{fischerf,weissg}@cs.tum.edu

Abstract

This paper proposes hierarchical reinforcement learning (RL) methods for communication in multiagent coordination problems modelled as Markov Decision Processes (MDPs).

To bridge the gap between the MDP view and the methods used to specify communication protocols in multiagent systems (using logical conditions and propositional message structure), we utilise *interaction frames* as powerful policy abstractions that can be combined with case-based reasoning techniques. Also, we exploit the fact that breaking communication processes down to manageable "chunks" of interaction sequences (as suggested by the interaction frames approach) naturally corresponds to methods proposed in the area of hierarchical RL.

The approach is illustrated and validated through experiments in a complex application domain which prove that it is capable of handling large state and action spaces.

1 Introduction

Communication is central to coordination and cooperation in multiagent systems (MASs) [13]. Unlike "physical" action that manipulates the environment in which agents are situated, communication may be used both in its own right (e.g. to exchange information) as well as to agree on the joint, coordinated execution of physical actions. The latter is possible because of the fact that the cost and hence the risk associated with communication is low compared to that of physical actions.

In the context of using communication to achieve coordinated behaviour, we can view agents' decision-making processes as *mediated* by communication, i.e. the utility of communicative decisions depends on the physical action(s) that result from them. This allows us to develop models of communicative decision-making in the framework of Markov Decision Processes (MDPs; e.g., see [7]), such that reinforcement learning (RL; e.g., see [11]) techniques are suited to derive an optimal *communication policy*. Learning such policies is essential to the design of adaptive agents in *open* MASs, where adherence to interaction and communication protocols cannot be taken for granted, and the need arises for agents to learn how a set of (possibly predefined) communication patterns can be applied strategically.

In this paper, we follow this interpretation and apply *hierarchical* RL methods [2] to the problem of communication learning. As we will see, these methods are well-suited

for communication-mediated multiagent MDPs, and we will support this intuition by experimental results in a complex domain. To induce hierarchical structure on the original MDP, powerful policy abstractions called *interaction frames* [10] are applied that allow for a generalisation over communication strategies. Since interaction frames are capable of handling speech-act-based [1] agent communication languages (ACLs) with propositional content, they bridge the gap between ACL and multiagent RL (MARL) research, so that this paper also contributes to the integration of knowledge-based agent and agent communication design with machine learning techniques.

However, we do *not* claim the suggested methods for learning strategic behaviour in communication to provide an optimal solution to the *overall* (i.e. physical+communicational) MARL problem. Rather, we develop hierarchical RL-based heuristics for dealing with a set of communication patterns specified (as is commonly done in MAS communication design) in terms of logical conditions and propositional message structure, thereby contributing to the agent design problem for communication learning problems.

The remainder of this paper is structured as follows. Section 2 gives a short overview of RL and the hierarchical RL framework of *options*. In section 3, we introduce interaction frames and present how this data structure can be combined with the RL techniques of section 2 to learn the effective use of a set of communication patterns. Experimental results in a complex application domain are reported on in section 4, and section 5 rounds off with some conclusions and an outlook on possible future work.

2 Reinforcement learning and the *options* framework

Standard RL is based on the MDP model of sequential decision processes. An MDP is defined by a finite set $\mathcal{S}$ of *states* and finite sets $\mathcal{A}_s$ of *admissible actions* for each state $s \in \mathcal{S}$. *Transition probabilities*

$$p^a_{ss'} = P(s_{t+1} = s' | s_t = s, a_t = a) \tag{1}$$

and *expected rewards*

$$r^a_s = E(r_{t+1} | s_t = s, a_t = a). \tag{2}$$

specify the system's behaviour if action $a \in \mathcal{A}_s$ is taken in state $s \in \mathcal{S}$ and time step t. In multiagent settings, the environment dynamics p includes the behaviour of other agents.

Agent behaviour is represented by means of a (stochastic) *policy* $\pi : \mathcal{S} \times \bigcup_{s \in \mathcal{S}} \mathcal{A}_s \rightarrow [0, 1]$, meaning that action a is executed with probability $\pi(s, a)$ whenever state s is perceived. According to the *expected discounted infinite-horizon reward maximisation* criterion which we follow here, an optimal policy π^* is one that maximises the expected sum $E(\sum_{j=0}^{\infty} \gamma^j r_{t+j})$ of discounted future rewards, where r_{t+j} is the reward obtained j steps in the future and $0 \leq \gamma < 1$ is a geometric discount factor.

Based on that, the objective of RL is to learn an optimal policy by sampling state transitions and rewards. Q-learning [12] solves this problem by learning the value $Q^*(s, a)$ of taking action a in state s and following π^* thereafter. For this, an approximation Q is updated according to

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha \left[r + \gamma \max_{a' \in \mathcal{A}_{s'}} Q(s', a') \right]$$

for a sampled transition from s to s' due to action a and with associated reward r. For a learning rate α appropriately decaying over time and using an exploration strategy which ensures that in the limit each action is executed infinitely often in each state, Q-learning can be shown to converge to Q^*. An optimal policy is then given by letting $\pi^*(s, a^*) = 1$ for $a^* = \arg\max_{a' \in \mathcal{A}_s} Q^*(s, a')$ and $\pi^*(s, a) = 0$ for all other a.

To allow for *temporal abstraction* in RL, we use the *options* framework [2] which is based on augmenting the sets $\mathcal{A}_s$ of admissible "primitive" actions by sets of so-called "options". An option is a triple $o = (\mathcal{I}, \pi, \beta)$ consisting of an input set $\mathcal{I} \subseteq \mathcal{S}$, a (stationary, stochastic) policy π over primitive actions, and a termination condition $\beta : \mathcal{S} \to [0, 1]$. Option o is admissible in a state s iff $s \in \mathcal{I}$. If invoked, o will behave according to π until it terminates stochastically with probability β (we assume that $\{s|\beta(s) < 1\} \subseteq \mathcal{I}$). $\mathcal{O}_s$ is used to denote the set of admissible options in state s, which may or may not include some or all of the primitive actions in $\mathcal{A}_s$. If an option's policy π is Markov, i.e. action probabilities depend solely on the state of the core MDP, the option itself is called a *Markov option*.

For greater flexibility with respect to action selection and to consider policies $\mu : \mathcal{S} \times \bigcup_{s \in \mathcal{S}} \mathcal{O}_s \to [0, 1]$ over options, however, so-called *semi-Markov options* are required. These build on the theory of *semi-Markov decision processes* (SMDPs; e.g., see [7]). In contrast to MDPs, the duration τ of an action within an SMDP is a random variable, such that temporally extended courses of action can be modelled. In the case of options, τ is the number of time steps from the invocation of an option to its termination.

Since the core MDP together with a set $\mathcal{O}$ of options constitutes an SMDP, SMDP learning methods can be used to learn an optimal policy over $\mathcal{O}$. In turn, with the core MDP serving as an explicit representation of the SMDP, intra-option policies can be evaluated and learned. The SMDP version of Q-learning applies the update equation

$$Q(s, o) \leftarrow (1 - \alpha)Q(s, o) + \alpha \left[r + \gamma^\tau \max_{o' \in \mathcal{O}_{s'}} Q(s', o') \right]$$

after option o has been running for τ time steps between s and s'. r denotes the cumulative discounted reward over this time, which could be computed as $r = r_1 + \gamma r_2 + \cdots + \gamma^{\tau-1} r_\tau$ from the individual rewards r_i. However, in an SMDP only the complete reward R obtained from executing o in s is known. Assuming an equal distribution of R over the τ steps yields

$$r = \sum_{i=1}^{\tau} \gamma^{i-1} \frac{R}{\tau} = \frac{\gamma^\tau - 1}{\gamma - 1} \cdot \frac{R}{\tau}.$$

$Q(s, o)$ can be shown to converge to $Q^*(s, o)$ for all $s \in \mathcal{S}$ and $o \in \mathcal{O}$ under conditions similar to those for conventional Q-learning.

3 Interaction frames

Interaction frames are a key concept of the abstract social reasoning architecture InFFrA proposed in [10]. There, they describe patterns of interactions that can be used strategically by knowledge-based agents to guide their communicative behaviour based on a reasoning process called *framing*.

For the scope of this paper, it suffices to look at (interaction) frames as *policy abstractions* (in the sense of MDP policies). This interpretation forms the basis of

a formal model of InFFrA called m²InFFrA (where the m² stands for "Markov-square" and hints at the underlying hierarchical two-level MDP view), details of which can be found in [9].

In m²InFFrA, a frame describes a set of two-party, discrete, turn-taking interaction *encounters* which can be thought of as conversations between two agents. A sequence of message patterns called *trajectory* specifies the surface structure of the encounters described by the frame, while a list of *substitutions* captures the values of variables in the trajectory in previously experienced interactions. Each substitution also corresponds to a set of logical *conditions* that were required for and/or precipitated by execution of the trajectory in the respective encounter. Finally, *trajectory occurrence* and *substitution occurrence* counters record the frequency with which the frame has occurred in the past. This leads to the following formal definition of m²InFFrA frames:

Definition 1

A frame *is a tuple* $F = (T, \Theta, C, h_T, h_\Theta)$*, where*

- $T = \langle p_1, p_2, \ldots, p_n \rangle$ *is a sequence of message patterns* $p_i \in \mathcal{M}$*, the* trajectory *of the frame,*
- $\Theta = \langle \vartheta_1, \ldots, \vartheta_m \rangle$ *is an ordered list of* variable substitutions*,*
- $C = \langle c_1, \ldots, c_m \rangle$ *is an ordered list of* condition sets*, such that* $c_j \in 2^{\mathcal{L}}$ *is the condition set relevant under substitution* ϑ_j*,*
- $h_T \in \mathbb{N}^{|T|}$ *is a* trajectory occurrence counter *list counting the occurrence of each prefix of the trajectory* T *in each of the previous encounters (even if the frame was not itself executed), and*
- $h_\Theta \in \mathbb{N}^{|\Theta|}$ *is a* substitution occurrence counter *list counting the occurrence of each member of the substitution list* Θ *in previous encounters (where the frame was actually executed).*

Thereby, $\mathcal{M}$ is a language of speech-act like message and action patterns of the form $\texttt{perf}(A, B, X)$ or $\texttt{do}(A, Ac)$. In the case of messages, `perf` is a performative symbol (`request`, `inform` etc.), A/B are agent identifiers or agent variables and X is the propositional content of the message taken from a logical language $\mathcal{L}$. In the case of physical actions with the special "performative" `do`, Ac is the action executed by A (an action has no recipient as it is assumed to be observable by any agent in the system). Both X and Ac may contain non-logical *substitution* variables that are used for generalisation purposes. We further use $\mathcal{M}_c \subset \mathcal{M}$ to denote the language of actual (ground) messages that agents use in communication (i.e. messages that do not contain variables other than "content variables" used in a logical sense).

Writing $T(F)$, $\Theta(F)$, etc. for functions that return the respective elements of a frame F, its semantics can informally be summed up as follows: The agent "owning" F has experienced $h_T(F)[1]$ encounters which started with a message matching the first element $m_1 = T(F)[1]$ of the trajectory. $h_T(F)[2]$ of these encounters continued with a message matching $m_2 = T(F)[2]$, and so on. Θ, h_Θ and C provide more information about specific past encounters: For $i \leq |\Theta|$, F represents $h_\Theta[i]$ past encounters matching $T(F)\Theta(F)[i]$, and $C(F)\Theta(F)[i]$ held during each of these encounters. Agents are assumed to maintain a knowledge base KB encoded in the same propositional language $\mathcal{L}$ that is used as a content language for messages.

From the standpoint of RL, ground instances of a frame can be seen as temporally extended policies that (like options) range over sequences of actions. Moreover, by virtue of generalisation over possible variable substitutions, a frame captures a whole set of such policies.

3.1 Frame-based options

After this general account, we will now describe how interaction frames can be integrated with the options framework. For this, we view agent communication as an MDP with a set $\mathcal{S}$ of states, which is derived using some kind of *state abstraction* that partitions the current knowledge base content KB and the perceived *encounter prefix* $w \in \mathcal{M}_c^*$ (the sequence of messages exchanged so far during an encounter) into equivalence classes denoted by $s_{(w,KB)}$. $\mathcal{A} = \mathcal{M}_c$ is used as the set of primitive actions.

For a frame F to induce an option $o_F \in \mathcal{O}$, $o = (\mathcal{I}_F, \pi_F, \beta_F)$, over the core MDP given by $\mathcal{S}$ and $\mathcal{A}$, we need to define $\mathcal{I}_F$, π_F and β_F appropriately based on F. Obviously, a frame can be selected iff there exists a substitution to enact it under. Thus, we have

$$\mathcal{I}_F = \left\{ s_{(w,KB)} \in \mathcal{S} \mid \Theta_{poss}(F, w, KB) \neq \emptyset \right\},$$

where $\Theta_{poss}(F, w, KB)$ is the set of substitutions F can be enacted under given encounter prefix w and knowledge base KB. Θ_{poss} can effectively be computed by unifying w with the appropriate prefix of $T(F)$ (which yields a "fixed" substitution $\vartheta_f(F, w)$) and restricting the variable bindings for the corresponding postfix $post(T(F), w)$ (to which ϑ_f has already been applied) to those executable under KB.

As for the termination of o_F, several reasons are imaginable:

1. o_F will definitely terminate if the end of the trajectory $T(F)$ has been reached.
2. o_F will also terminate if $T(F)$ no longer matches the encounter prefix w.
3. Changes to the knowledge base may prohibit the execution of the remaining steps of F. As above, this is verified by checking whether $\Theta_{poss}(F, w, KB) = \emptyset$.
4. The remaining steps of F may be rendered undesirable due to changes to the agent's knowledge base.
5. Actions by the peer may prohibit execution of the frame under the most desirable substitution.

Items 2, 3 and 4 concern the *validity*, *adequacy* and *desirability* of F, respectively [10]. Item 5 can be viewed as concerning the desirability of F as well as the validity of the most desirable substitution.

If we assume that desirability (conditions 4 and 5) corresponds to the profit the agent will obtain from executing a message/action sequence w under knowledge base KB which it can estimate using a utility function $u : \mathcal{M}_c^* \times KB \to \mathbb{R}$, then a boolean desirability criterion $\delta(F, w, KB)$ can be defined which determines desirability-based option termination:

$$\beta_F(s_{(w,KB)}) = \begin{cases} 1 & \text{if } \Theta_{poss}(F, w, KB) = \emptyset \\ & \text{or } \delta(F, w, KB) \\ 0 & \text{otherwise.} \end{cases}$$

As [2] points out, optimal policies over the set of available options are in general suboptimal policies of the core MDP, if not all primitive actions $a \in \mathcal{A}_s$ remain admissible in s. This is obvious for the special case of frame-based options, since a frame's expected reward may change during the enactment, rendering another frame more desirable. As we have already said, we are willing to accept this drawback for the sake of complexity reduction, all the more in the domain of multiagent interaction another benefit may arise: If agents accept frames as an established means of interaction and follow them normatively to a certain extent rather than constantly driving for optimal actions, this will make them more comprehensible and dependable, thereby reducing the contingency inherent in interactions. As a result, agents should adhere to a frame as long as possible.

What now remains to be specified is the intra-option policy π_F corresponding to a frame F. From a rational actor's point of view, the agent should take the best possible action according to its utility function u, considering the restrictions imposed by the active frame. This yields the (greedy, deterministic) *enactment policy*

$$\pi_F(s_{(w,KB)}, m) = \begin{cases} 1 & \text{if } m = m^*(F, w, KB) \\ 0 & \text{otherwise,} \end{cases}$$

where $m^*(F, w, KB)$ is the optimal action given encounter prefix w and knowledge base KB and restricted by frame F. A concrete definition of m^* will be given in the following section.

To ensure convergence of Q-learning, we can add *Boltzmann exploration* to obtain a stochastic frame selection criterion with a temperature T that decreases over time:

$$P(F|w, KB) = \frac{e^{Q(s_{(w,KB)}, o_F)/T}}{\sum_{\Theta_{poss}(F', w, KB) \neq \emptyset} e^{Q(s_{(w,KB)}, o_{F'})/T}}$$

One might argue that by using single-agent RL instead of "real" MARL like Markov games, convergence would only be achieved if peer agents followed a stationary policy. This is not the case however, since our approach allows different policies to be learned for different policies of the peer, so these are virtually stationary within a current encounter.

3.2 Frame enactment

To determine the optimal action m^* we should select *within* a frame, we apply expected utility maximisation within the temporal scope of the remaining trajectory steps, since, in general, the postfix of $T(F)$ with respect to w can contain unbound variables so that the utility of its execution is not ex ante deterministic.

From the framing view, both the agent itself and the peer it is interacting with have the freedom to substitute concrete values for free variables that occur first in one of their trajectory steps. We will write Θ_s and Θ_p for the sets of possible substitutions the agent and the peer can apply respectively and ϑ_s and ϑ_p for specific elements of these sets. Then, the expected utility of executing the remaining steps of $T(F)$ is given by

$$E[u(\vartheta_s|F, w, KB)] \;=\; \sum_{\vartheta_p \in \Theta_p} u_\gamma(post(T(F), w)\vartheta_s\vartheta_p, KB) \;\cdot\; P(\vartheta_p|\vartheta_s, F, w),$$

where $post(T(F), w)$ again denotes the postfix of $T(F)$ with respect to prefix w, $u_\gamma(w, KB)$ is the discounted utility of executing a message sequence w with initial

knowledge base KB and $P(\vartheta_p|\vartheta_s, F, w)$ is the probability with which the peer will conditionally choose a substitution ϑ_p depending on the agent's own choice ϑ_s. Based on that, the optimal action is given by

$$m^*(F, w, KB) = T(F)[|w| + 1]\vartheta^*(F, w, KB),$$

where

$$\vartheta^*(F, w, KB) = \arg\max_{\vartheta_s \in \Theta_s} E[u(\vartheta_s|F, w, KB)].$$

To compute the probability $P(\vartheta_p|\vartheta_s, F, w)$ in accordance with the model provided by the frame F, we will compare the (projected) message sequence of the present encounter with those of the (past) encounters stored in F.

According to the consequentialist and empirical view of communication, the future probability for the occurrence of any message sequence should be equal to the frequency with which it has been observed in the past. However, $T(F)$ can be very abstract, an it is unlikely that all the past cases stored in F are equally relevant for every new encounter prefix that matches $T(F)$. Intuition suggests that this relevance should be expressed using some notion of *similarity* between message patterns in the vein of *case-based reasoning* [6]. To formally capture this notion, we introduce a real-valued *similarity measure* $\sigma : \mathcal{M}^* \times \mathcal{M}^* \to [0, 1]$ on sequences of messages, allowing us to compare the (perceived) encounter prefix with the past cases stored in a frame. In general, the definition of σ will be domain-dependant. A very simple default choice that proves viable in many cases is to define sequence similarity recursively as the average pairwise similarity of sequence elements and their arguments. At the term/operator level, a strict equality criterion can be applied while assigning a similarity of 1 to term/variable and variable/variable pairs.

Based on this similarity measure on message sequences, the similarity of a substitution ϑ to a frame F can be defined as

$$\sigma(\vartheta, F) \;=\; \sum_{i=1}^{|\Theta(F)|} \sigma\big(T(F)\vartheta, T(F)\Theta(F)[i]\big) \;\cdot\; h_\Theta(F)[i] \;\cdot\; c\big(C(F)[i], \vartheta, KB\big).$$

where $c(C, \vartheta, KB)$ is 1 if $C\vartheta$ holds under KB and 0 else (where obvious from the context, we omit KB for readability). The probability that a frame F is enacted under a specific substitution ϑ is then computed as the similarity of ϑ to F relative to all substitutions in Θ_{poss}, i.e.

$$P(\vartheta|F, w) = \begin{cases} \lambda \cdot \sigma(\vartheta, F) & \text{if } \vartheta \in \Theta_{poss}(F, w, KB) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

for a normalisation constant λ.

To determine $P(\vartheta_p|\vartheta_s, F, w)$ we can use the Bayesian product rule

$$P(\vartheta_p \wedge \vartheta_s|F, w) = P(\vartheta_p|\vartheta_s, F, w) \cdot P(\vartheta_s|F, w),$$

where $\vartheta_p \wedge \vartheta_s$ denotes the event of the peer selecting $\vartheta_p \in \Theta_p$ and the agent selecting $\vartheta_s \in \Theta_s$, such that F is enacted under the complete substitution that results from combining the fixed substitution ϑ_f with ϑ_p and ϑ_s.

On the other hand, the probability that the agent has previously chosen substitution ϑ_s is given by the sum of the probabilities for the occurrence of complete substitutions

that ϑ_s is part of, such that

$$P(\vartheta_p|\vartheta_s, F, w) = \frac{P(\vartheta_p \wedge \vartheta_s|F, w)}{P(\vartheta_s|F, w)} = \frac{P(\vartheta_f(F, w)\vartheta_s\vartheta_p|F, w)}{\sum_\vartheta P(\vartheta_f(F, w)\vartheta_s\vartheta|F, w))}.$$

Applying equation 3 to both numerator and denominator finally yields

$$P(\vartheta_p|\vartheta_s, F, w) = \frac{\sigma(\vartheta_f(F, w)\vartheta_s\vartheta_p, F)}{\sum_\vartheta \sigma(\vartheta_f(F, w)\vartheta_s\vartheta, F)},$$

provided that $\vartheta_f(F, w)\vartheta_s\vartheta_p \in \Theta_{poss}(F, w, KB)$ (observe that the denominator is constant in ϑ_p and does not need to be computed to determine $\vartheta^*(F, w, KB)$).

To sum up, a frame F is enacted by executing the next step of the trajectory $T(F)$ under the substitution that promises the highest expected utility for the complete trajectory suffix, while computation of the occurrence probability for each substitution is based solely on its similarity to the past cases stored in F. In [4], this form of reasoning about communication as well as the underlying concept of empirical communication semantics are examined with greater detail.

4 Experimental results

The learning approach presented in the previous sections has been tested in the multiagent-based link exchange system LIESON. In this system, agents representing Web sites engage in communication to negotiate over mutual linkage with the end of increasing the popularity of one's own site and that of other preferred sites.

Available physical actions in this domain are the addition and deletion of numerically rated links originating from one's own site and the modification of ratings (where the probability of attracting more traffic through a link depends on the rating value).

LIESON provides a highly dynamic and complex interaction testbed for the following reasons:

- Agents only have a partial and incomplete view of the link network. In particular, agents engage in non-communicative goal-oriented action in between encounters, so that the link network (and hence the agents' utility situation) may change while a conversation is unfolding.
- The number of possible link configurations is vast, and agents can only predict possible utilities for a very limited number of hypothetical future layouts.
- There is no notion of commitment – agents choose frames in a self-interested way and may or may not execute the physical actions that result from them. Also, they may undo their effects later on.

LIESON agents consist of a non-social BDI [8] reasoning kernel that projects future link network configurations and prioritises goals according to utility considerations. If these goals involve actions that have to be executed by other agents, the m²InFFrA component starts a framing process which runs until the goal of communication has been achieved or no adequate frame can be found. We report on experiments in which these agents were equipped with frames with the following six trajectories:
`request(A,B,X)`→`accept(B,A,X)`→`confirm(A,B,X)`→`do(B,X)`
`request(A,B,X)`→`propose(B,A,Y)`→`accept(A,B,Y)`→`do(B,Y)`

`request(A,B,X)`→`prop-also(B,A,Y)`→`accept(A,B,Y)`→`do(B,X)`→`do(A,Y)`
`request(A,B,X)`→`reject(B,A,X)`
`request(A,B,X)`→`propose(B,A,Y)`→`reject(B,A,Y)`
`request(A,B,X)`→`prop-also(B,A,Y)`→`reject(B,A,Y)`

The first three frames allow for `accept`ing to perform a `request`ed action X, making a counter-`propose`al in which Y is suggested instead of X, or using `prop-also` to suggest that B executes X if A agrees to execute Y. The last three frames can be used to explicitly `reject` a request or proposal. In that, X and Y are link modification actions; each message is available in every state and incurs a cost that is almost negligible compared to the utilities gained or lost through linkage actions (yet high enough to ensure no conversation goes on forever). Also, agents can always send a `stop` action to indicate that they terminate an encounter if they cannot find a suitable frame.

After termination, encounters are stored in the frame from which they have originated. For example, agent a_1 would store the encounter $request(a_1, a_2, add(a_2, a_1, 2)) \rightarrow reject(a_1, a_2, add(a_2, a_1, 2))$ by adding a substitution $[A/a_1, B/a_2, X/add(a_2, a_1, 2)]$ to the respective frame together with an automatically generated list of conditions that were required for physical action execution.

As state abstraction, we represent the physical actions referred to in an encounter using statements of the form $\{\uparrow|\downarrow\}(\{I, R\}, \{I, R, T\}, \{+, -, ?\})$. $\uparrow$ and $\downarrow$ stand for a positive/negative link modification (i.e. addition/deletion of a link or an increase/decrease of its rating value), I/R for the initiator/responder of the encounter, T for a third party; $+/-/?$ indicates whether the (learning) agent likes/dislikes/doesn't know the target site of the link modification. For example, if a_1 and a_2 talk about $do(a_1, deleteLink(a_1, a_3))$ in an encounter initiated by a_1 (while the learning agent a_2 is the responder and likes a_3's site) this is abstracted to $\downarrow(I, T, +)$. If in the same conversation a_2 suggests to modify his own link toward a_1 (whom he does not like) from a rating value of 1 to 3, the state (viz *subject*) of the encounter becomes $\{\downarrow(I, T, +), \uparrow(R, I, -)\}$. The intuition behind this state abstraction method is to capture, in a generalised form, the *goal* of the conversation that can currently be realised while at the same time reducing the state space to a reasonable size.

Figure 1 shows a comparison for a system with ten agents with an identical profile of private ratings (preferences) towards other agents (both plots show the performance of the best and the worst agent in the group as well as the average utility over all agents, averaged over ten independent runs). In the first plot, agents employ BDI reasoning and additionally send requests to others whenever they favour execution of someone else's action according to their BDI queue. These requests are then enqueued by the recipient as if he had "thought of" executing the respective action himself. Thus, it depends on the recipient's goal queue and on his utility considerations whether the request will be honoured or not. As one can see, after a certain amount of time agents no longer execute any of the actions requested by others, and cannot find any profitable action to execute themselves, either. The system converges to a stable state.

The second plot shows the results of a simulation with the same setup as above but using m^{2}InFFrA agents. Again, agents issue requests whenever they identify that someone else could do something useful. After this initial message, the framing procedure takes over. Quite clearly, despite the fact that there is a greater variation in maximal/minimal/average agent utility, the average and the best agent perform significantly better than in the BDI case, while the weakest agent performs just as good as in the BDI case on the average.

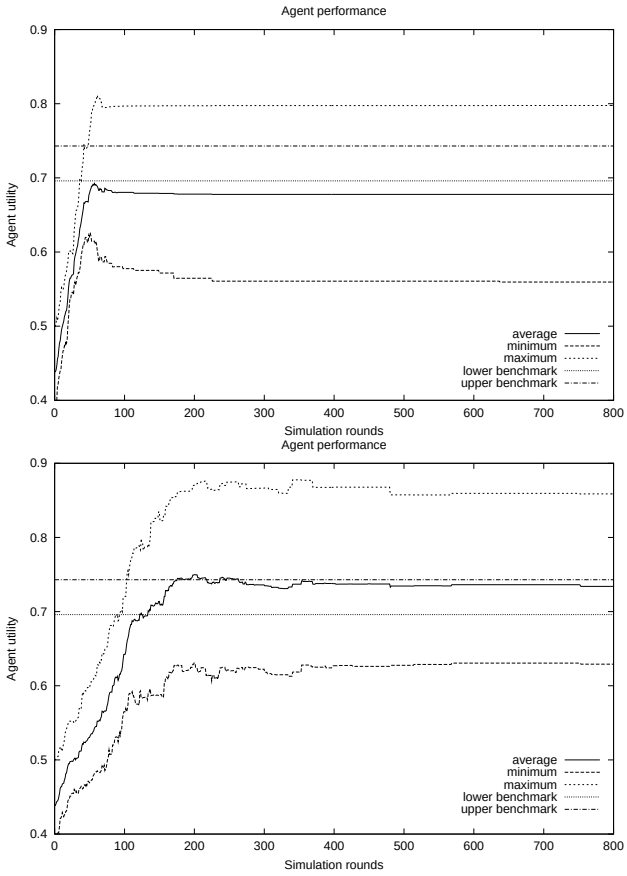

Figure 1: Performance plots.

While on first glance the performance of m²InFFrA agents might not be strikingly different from that of plain BDI agents, it should be noted that the results establish a lower bound on the performance gained by using m²InFFrA. In environments with (possibly) non-benevolent peer agents showing non-stationary behaviour, the performance of agents using prespecified communication protocols or assuming a fixed semantics of communication can become arbitrarily bad, while m²InFFrA includes the ability to learn what to expect from a given peer in a specific interaction situation. To allow for any quantitative statements, however, additional experiments will be necessary.

More interesting still is that the average utility lies within the range of the two horizontal lines in the plot. These denote the average utilities for two very interesting linkage configurations: the lower of the two corresponds to a fully connected linkage graph, in which each agent (honestly) displays the ratings of his out-links, i.e. reveals his true opinions about others. The slightly higher utility shown by the upper line is attained if agents do not lay any links toward agents they dislike. It is an interesting property of the utility function used in LIESON, that being "politically correct" is slightly better than being honest. The fact that agent utilities evolve around these benchmarks indicates that they truly strive to make strategic communication moves and to exploit the advantages of concealing certain beliefs.

5 Conclusions

In this paper, we have proposed hierarchical RL methods for learning communication strategies in multiagent coordination, using *interaction frames* as a rich representation for policy abstractions.

We have formally defined frames in the m²InFFrA framework as sets of encounter patterns supplemented with logical conditions, variable substitutions and occurrence counters. By virtue of the options framework, frames have been re-interpreted as temporal abstractions in the sense of hierarchical RL. Also, by applying similarity criteria, they can be seen as case abstractions in terms of case-based reasoning. We have defined a two-level hierarchical decision-making apparatus for learning and reasoning with frames and underlined its usefulness through experiments in a complex multiagent domain.

A major advantage of our approach is that it combines the decision-theoretic power of RL models with the knowledge-based aspects of symbolic agent communication, interaction protocols and ACL research in general. Compared to other approaches that use hierarchical RL to learn communication policies (e.g. [5]), it is much closer to the relational and situated nature of communication and interaction in a MAS and allows for an explicit representation of first-order message content and logical conditions. It is this aspect that makes rational action and learning possible for high-level agent architectures that employ logical reasoning.

Future work includes investigations into how interaction frames can be constructed from scratch (first steps in this direction concerning the concatenation of frames have been described in [3]). Developing the theory of *hierarchical options* (built around a policy over options) [2] into "meta-frames" that allow for an online combination of different interaction patterns and subgoals seems to be a promising idea for the construction of frames for longer-term interactions. Also, the issue of some general form of state abstraction is still largely unresolved and deserves our attention in the future.

References

[1] John L. Austin. *How to do things with Words*. Clarendon Press, 1962.

[2] Andrew G. Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):41–77, 2003.

[3] Felix Fischer. Frame-based learning and generalisation for multiagent communication. Diploma Thesis. Department of Informatics, Technical University of Munich, 2003.

[4] Felix Fischer and Michael Rovatsos. Reasoning about communication: A practical approach based on empirical semantics. In *Proceedings of the 8th International Workshop on Cooperative Information Agents (CIA-04)*, volume 3191 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2004.

[5] Mohammad Ghavamzadeh and Sridhar Mahadevan. Learning to communicate and act in cooperative multiagent systems using hierarchical reinforcement learning. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, New York, NY, 2004.

[6] Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Francisco, CA, 1993.

[7] Martin L. Puterman. *Markov Decision Processes*. John Wiley & Sons, New York, NY, 1994.

[8] Anand S. Rao and Michael P. Georgeff. An abstract architecture for rational agents. In W. Swartout C. Rich and B. Nebel, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-92)*, pages 439–449, 1992.

[9] Michael Rovatsos. *Computational Interaction Frames*. PhD thesis, Department of Informatics, Technical University of Munich, under review, 2004.

[10] Michael Rovatsos, Gerhard Weiß, and Marco Wolf. An Approach to the Analysis and Design of Multiagent Systems based on Interaction Frames. In Maria Gini, Toru Ishida, Cristiano Castelfranchi, and W. Lewis Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, Bologna, Italy, 2002. ACM Press.

[11] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning. An Introduction*. MIT Press/A Bradford Book, Cambridge, MA, 1998.

[12] C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[13] Gerhard Weiß, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, 1999.