

An Integrated Framework for Adaptive Reasoning About Conversation Patterns

Michael Rovatsos
School of Informatics
The University of Edinburgh
Edinburgh EH8 9LE
United Kingdom
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

We present an integrated approach for reasoning about and learning conversation patterns in multiagent communication. The approach is based on the assumption that information about the communication language and protocols available in a multiagent system is provided in the form of dialogue sequence patterns, possibly tagged with logical conditions and instance information. We describe an integrated social reasoning architecture that is capable of (i) processing such patterns, (ii) making communication decisions in a boundedly rational way, and (iii) learning patterns and their strategic application from observation. Our method combines decision-theoretic utility maximisation, case-based reasoning methods, hierarchical reinforcement learning and cluster validation techniques to yield a comprehensive model of communicative decision-making and learning that can be practically implemented. The adequacy of the approach is illustrated through experimental results in complex negotiation scenarios.

1. Introduction

Compared to the long-established areas of interaction protocol and agent communication language (ACL) research [7, 3], the development of agent architectures suitable for dealing with provided communication mechanisms in practical terms has received fairly little attention.

As yet, there exists no uniform framework for defining the interface between the inter-agent communication layer and intra-agent reasoning, i.e. to define how specifications of interaction protocols and communication semantics influence agent rationality or, in turn, are influenced themselves by agents' rational decision-making processes.

Moreover, there is a growing concern that most specification methods for ACLs and interaction protocols do not provide sufficient guidance as to which part of the *semantics* of communication should be specified at a supra-agent level and which part of them is only a result of agents' mental processing and cannot be captured without knowledge of their internal design. Clearly, concentrating on either of these two sides may lead to over-constraining agent autonomy (as agents would merely "execute" centralised communication procedures that directly modify their internal states) or to uncertainty about the consequences of communication (e.g. in terms of adherence to previously created commitments) and loss of social structure altogether.

In this paper, we attempt to tackle this problem from a very pragmatic perspective. We make very weak assumptions regarding the specification method used for defining the available means of communication in a multiagent system (MAS), namely that it provides (i) descriptions of the surface structure of communication processes (in the simplest case, traces of possible message and action sequences in agent conversations) that are tied to (ii) some form of logical constraints (represented in a tractable logical language, if they are to be used by reasoning agents). We refer to such pairs of surface structure and logical constraints as *conversation patterns* in the following.

On the agent side, we shall assume a rather simple decision-theoretic design by which agents are striving for long-term utility maximisation. We assume that they dispose of a utility estimation method which helps them prioritise (their own or others') actions based on the immediate utility provided by these actions. Also, we endow endowed with knowledge about a set of conversation patterns in the above sense to guide their communication behaviour. In an autonomy-respecting fashion, however, we do *not* assume that agents will always adhere to these patterns, i.e. in principle they can violate existing patterns both in terms of using them even if the respective logical conditions are not

met as well as deviating from the message sequences specified by the patterns.

In the remaining sections, we define an architecture for reasoning about such conversation patterns (section 2), applying them in communication situations (section 3), and adapting conceptions of these patterns from experience (section 4). We illustrate its applicability with examples and experimental results in section 5. Section 6 concludes.

2. Conversation Patterns and Frames

Although there exists a multitude of methods for specifying ACL semantics and interaction protocols, their greatest common denominator is that they describe the *surface structure* of dialogues (i.e. a set of admissible message sequences) and logical *constraints* for the applicability of these message sequences (which may include statements about environmental conditions, mental states of the participating agents, the state of commitment stores, etc.). In the most simplistic case, these structure/constraint pairs may be represented as a set of combinations of a conversation trace and a set of conditions. For example,

$$\langle \text{request}(a, b, \text{pay}(\$100)) \rightarrow \text{do}(b, \text{pay}(\$100)), \quad (1) \\ \{ \text{can}(b, \text{pay}(\$100)) \} \rangle$$

can be used to express that a request of agent a is followed by an action if the requestee b is able to execute the action, i.e. pay a an amount of \$100.

The question that serves as a point of departure for our research is how we can build agents capable of processing a set of such (conditioned) conversation patterns in a goal-oriented and adaptive fashion, given that the reliability of these specification is contingent on other's (and the agent's own) adherence to their prescriptive content.

For developing such an agent design, the social reasoning meta-architecture InFFrA (Interaction Frames and Framing Architecture) [13] is ideally suited. In InFFrA, communication processes are conceptually described by so-called *interaction frames*. Each of these interaction frames represents a particular category of interactions by describing attributes of the respective category. These attributes concern the

- *trajectory*, i.e. surface structure of the interaction,
- *roles* of interacting parties and *relationships* between them
- *context* and *belief* conditions that constrain the applicability of the frame to certain situations.

Using such frames as central data models, InFFrA describes a reasoning cycle (called *framing*) which includes

- *situation interpretation* to match the currently perceived interaction against existing frame conceptions

(that are assumed to be contained in a locally maintained *frame repository*),

- *frame assessment* to determine whether the currently *active frame* should be retained (and guide the agent's interaction behaviour in the current situation),
- *re-framing*, i.e. selection of a better alternative or adaptation of existing frame conceptions, and
- *frame enactment* which results in applying the active frame as a normative model of the current interaction situation in one's decision making.

In $m^2\text{InFFrA}$, which is a concrete instance of InFFrA suggested by Fischer and Rovatsos [5] frame trajectories are modelled as linear sequences of message patterns (i.e. speech-act like messages which may contain variables for the participating agents and content elements) which represent discrete, turn-taking communicative encounters between two agents. These trajectories are combined with sets of variable substitutions and logical conditions which provide instance and applicability information. Also, the frequencies of these substitutions and the frequency with which prefixes of the perceived conversation sequence matched a frame in the past are recorded inside the frame.

Formally, an $m^2\text{InFFrA}$ frame is a tuple $F = (T, \Theta, C, h, h_\Theta)$, where

- $T = \langle p_1, p_2, \dots, p_n \rangle$ is a sequence of message patterns $p_i \in \mathcal{M}$, the *trajectory* of the frame,
- $\Theta = \langle \vartheta_1, \dots, \vartheta_m \rangle$ is an ordered list of *variable substitutions*,
- $C = \langle c_1, \dots, c_m \rangle$ 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 previous encounters, 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.

In this definition, \mathcal{M} is a language of speech-act like message and action patterns of the form $\text{perf}(A, B, X)$ or $\text{do}(A, Ac)$. In the case of messages (i.e. exchanged textual signals), perf is a performative symbol (`request`, `inform`, etc.), A and B are agent identifiers or agent variables and X is the content of the message taken from a first-order language \mathcal{L} . In the case of "physical" actions (i.e. actions that manipulate the physical environment) with the special "performative" `do`, Ac is the action executed by A (a physical 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 ground (up to variable identifiers of the content language) messages that agents use in communication. To illustrate the semantics of a frame, consider the following example¹:

$$F = \left\langle \left\langle \begin{array}{l} \xrightarrow{5} \text{request}(A, B, X) \xrightarrow{3} \text{do}(B, X), \\ \{ \text{can}(B, X) \}, \\ \{ \text{can}(B, \text{pay}(S)) \} \end{array} \right\rangle, \right. \\ \left. \left\langle \begin{array}{l} \xrightarrow{2} \langle [A/a], [B/b], [X/\text{pay}(\$100)] \rangle, \\ \xrightarrow{1} \langle [A/b], [B/a], [X/\text{pay}(S)] \rangle \end{array} \right\rangle \right\rangle \quad (2)$$

This frame reflects the following interaction experience: A asked B five times to perform (physical) action X , out of which B actually did so in three instances. In two of successful instances, it was a who asked and b who headed the request, and the action was to pay \$100. In both cases, $\text{can}(b, \text{pay}(\$100))$ held true. In the third case, roles were swapped between a and b and the amount S remains unspecified (which does not mean that it did not have a concrete value, but that this was abstracted away in the frame).

What is important to note about m^2lnFFrA frames in contrast to general conversation patterns (cf. example 1) is that they allow for storing *empirical* information about past instances of conversations that followed a certain pattern and also to distinguish between different sets of conditions that held during these *enactments* of a frame. Thus, apart from surface structure and logical constraints, frames also allow for storing *instance information*.

3. Applying Interaction Frames

The ability of frames to capture instance information provides agents with a facility to reason about communication semantics in an adaptive fashion. In accordance with the *empirical semantics* view [12] that considers the meaning of communication as a function of its consequences as experienced through the eyes of a subjective observer, agents can adapt existing frame conceptions with new observations of encounters and project past regularities into the future. As we will see, this may help improve their strategic communication abilities decisively in *open* systems, in which agents may or may not obey pre-defined conversation patterns.

¹ For ease of presentation, we will write $T(F)$, $C(F)$, etc. to denote the respective elements of a frame F and use the compact notation $\langle T_h(F), C(F), \Theta_h(F) \rangle$ instead of $\langle T, \Theta, C, h, h_\Theta \rangle$, where $T_h(F) = \xrightarrow{h(F)[1]} p_1 \xrightarrow{h(F)[2]} p_2 \dots \xrightarrow{h(F)[n]} p_n$ and $\Theta_h(F)[i] = \xrightarrow{h_\Theta(F)[i]} \Theta(F)[i]$.

3.1. Frame semantics

According to the *probabilistic interpretation* of frame semantics that underlies the m^2lnFFrA model, the semantics of a set of frames $\mathcal{F} = \{F_1, \dots, F_n\}$ is as follows: Given an *encounter prefix* $w \in \mathcal{M}_c^*$, i.e. a sequence of messages already uttered in the current encounter (possibly the empty sequence) and the *knowledge base* $KB \in 2^{\mathcal{C}}$ of beliefs² the reasoning agent currently holds, we can compute the set of possible *continuations* $w' \in \mathcal{M}_c^*$ by

1. filtering out all those frames whose trajectories do not prefix-match w ,
2. considering the postfixes of w in the remaining frames under the remaining possible substitutions (given that w has already committed certain variables to instance values), and
3. applying those substitutions whose corresponding condition sets are fulfilled under KB .

For each of these possible continuations, we can then compute a *continuation probability* by virtue of similarity, frequency and relevance considerations. The resulting probability distribution over continuations w' is the *semantics* of w under \mathcal{F} .

Formally, let $\vartheta_f(F, w) = \text{unifier}(w, T(F)[1:|w|])$ the most general unifier of w and the corresponding trajectory prefix $T(F)[1:|w|]$ of F (where $\text{unifier}(\cdot, \cdot)$ returns the most general unifier for two message patterns or sequences thereof, or \perp if they cannot be unified). This can be used to define

$$\Theta_{\text{poss}}(F, KB, w) = \{ \vartheta | \exists \vartheta'. \vartheta = \vartheta_f(F, w) \vartheta' \\ \wedge \exists i. KB \models C[i] \vartheta \}$$

as the set of *possible substitutions* under frame F , beliefs KB , and conversation prefix w ($S\vartheta$ denotes application of substitution ϑ to a (set or list of) logical formula(e) or message(s) S depending on the context). In other words, Θ_{poss} is the set of substitutions that are extensions of ϑ_f for which at least one condition in $C(F)$ is satisfied.

This allows us to specify which continuations w' of w should be expected to occur with non-zero probability according to F under KB , namely exactly those that result from applying any possible substitution to the postfix of the trajectory. Now the question arises how the quantities of these probabilities should be determined, i.e. what the probability $P(\vartheta|F, w)$ under a frame F and a given prefix w should be.

Using a *similarity measure* $\sigma(\cdot, \cdot)$ defined on message pattern sequences as used in the area of *case-based reasoning* [6], we can determine the similarity of any possi-

² For notational convenience, we assume that knowledge bases use the same logical language as is used in the content language of messages.

ble substitution to a frame taking into account the frequencies of previous cases and the relevance of their corresponding condition sets in a single frame by defining

$$\sigma(\vartheta, F) = \sum_{i=1}^{|\Theta(F)|} \left(\underbrace{\sigma(T(F)\vartheta, T(F)\Theta(F)[i])}_{\text{similarity}} \cdot \underbrace{h_{\Theta(F)}[i]}_{\text{frequency}} \right) \cdot \underbrace{c_i(F, \vartheta, KB)}_{\text{relevance}}$$

to assess to which extent ϑ is “applicable” to F .³ This can be used to assign a probability

$$P(\vartheta|F, w) \propto \sigma(\vartheta, F) \quad (3)$$

to all $\vartheta \in \Theta_{\text{poss}}(F, KB, w)$ and a probability of zero to all other ϑ .

Finally, this allows for computing continuation probabilities⁴ for w over the entire set of frames \mathcal{F} as

$$P(w'|w) = \sum_{F \in \mathcal{F}, ww' = T(F)\vartheta} P(\vartheta|F, w)P(F|w) \quad (4)$$

where $P(F|w)$ is the number $h(F)[|h(F)|]$ of successful completions of F normalised over all frames that prefix-match w .

3.2. Decision making with frames

In the introductory section, we assumed a decision-theoretic agent design by which agents strive for long-term expected utility maximisation. Further assuming that agents are equipped with a *utility estimate* $u(w, KB) \in \mathbb{R}$ that allows them to assess the usefulness of a message (and action) sequence w in belief state KB (e.g. by assigning substantial positive/negative utility values to do-messages that represent physical actions in the environment and a small negative utility to messages to express the communication cost incurred by them), they could in principle use continuation probabilities (cf. equation 4) to make utility-maximising decisions in each reasoning cycle.

However, this contradicts the intuition behind InFFrA which relies on breaking down the whole network of expectations into manageable “chunks” i.e. frames, and is also counter-intuitive with respect to the ways in which conversation pattern specifications are usually available in most MASs (different protocols are specified for different purposes, and need not all be constantly be reasoned over at

³ An easy way to define $c_i(F, \vartheta, KB)$ is to assign a value of 1 to this quantity if $KB \models C(F)[i]\Theta(F)[i]$ and 0, else.

⁴ Note that $P(w'|w)$ will not be a real probability distribution over \mathcal{M}_c^* unless we extend it to those w' for which no frame postfix caters. This can be done, for example, by assigning small uniform probabilities to all these w' .

the same time while engaging in a particular kind of interaction).

Instead we will assume in a boundedly rational fashion, and in accordance with the general InFFrA reasoning cycle, that an agent only activates a single frame at a time within which it searches for an optimal action while engaging in a communicative encounter. It will only *re-frame* if this is required because the current frame is not applicable anymore or does not seem desirable in terms of utility expectations.

The decision-making procedure of m^2 InFFrA can be outlined as follows:

1. If a message m is received, update the encounter prefix: $w \leftarrow wm$; if no encounter is running, consider starting one
2. If no active frame F has been selected, go to 8.
3. *Validity check*: If $|T(F)| = w$, go to 7.; if $unifier(T(F)[1 : |w|], w) = \perp$, go to 8.
4. *Adequacy check*: If $\Theta_{\text{poss}}(F, w, KB) = \emptyset$, go to 8
5. Compute the expected utility for each *own substitution* ϑ_s :

$$E[u(\vartheta_s, F, w, KB)] = \sum_{\vartheta_p} \left(u(\text{postfix}(T(F), w)\vartheta_s\vartheta_p, KB) \cdot P(\vartheta_p|\vartheta_s, F, w) \right)$$

Desirability check: Determine the optimal substitution

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

If $u(\text{postfix}(T(F), w)\vartheta^*\vartheta_p, KB) < b$, go to 8.

6. Perform $m^* = T(F)[|w| + 1]\vartheta^*$, update $w \leftarrow wm$
7. Wait for next message until deadline; after that, terminate encounter
8. *Re-frame* to select F , then go to 3.

The framing reasoning cycle is bracketed by steps 1 and 7 which cater for initiating encounters and ending them if no more messages are received (if the other agent is not replying at all when he is expected to, and to make sure we process additional messages sent by the other party after we considered the encounter completed (coming from step 3)).

We assume that encounter initiation on the side of the agent is spawned by some sub-social reasoning layer, e.g. a BDI [10] engine, which determines whether and about what to converse with whom, depending on the possibility of furthering some local goal through such interaction.

Steps 3, 4 and 5 are used to evaluate the usefulness of the currently active frame F . Steps 3 and 4 are straightforward: If the frame has been completed, if it does not prefix-match the encounter prefix w or if $\Theta_{\text{poss}}(F, w, KB)$ becomes empty, F cannot be used any longer.

For step 5, we have to assess the expected utility $E[u(\vartheta_s, F, w, KB)]$ of any ‘‘own’’ substitution ϑ_s . To this end, we have to conduct an adversarial search over substitutions jointly determined by the agent and her opponent, as each of the two agents commits certain variables to concrete values in their turn-taking moves. Using the definition of $\sigma(\vartheta, F)$ and equation 3, we can estimate the probability for opponent substitution ϑ_p in the remaining steps as

$$\begin{aligned} P(\vartheta_p|\vartheta_s, F, w) &= \frac{P(\vartheta_p \wedge \vartheta_s|F, w)}{P(\vartheta_s|F, w)} \\ &= \frac{\sigma(\vartheta_f(F, w)\vartheta_s\vartheta_p, F)}{\sum_{\vartheta} \sigma(\vartheta_f(F, w)\vartheta_s\vartheta, F)} \end{aligned}$$

where $\vartheta_p \wedge \vartheta_s$ denotes the event of the peer choosing ϑ_p and the reasoning agent choosing ϑ_p after having committed to the fixed substitution $\vartheta_f(F, w)$, so that the final ‘‘joint’’ substitution will be $\vartheta_f(F, w)\vartheta_s\vartheta_p$.

With this, we can use u to compute the utility of the postfix that results from $T(F)$ and prefix w (which implies application of $\vartheta_f(F, w)$) with ϑ_p and ϑ_s applied to obtain a ground sequence of actions and messages still to be executed along the remaining trajectory. If the utility of the postfix under the optimal substitution ϑ^* is below some threshold b , we pick a new frame. Otherwise, we perform the next step m^* along the trajectory of F and enter the next reasoning iteration.

It should be mentioned that we have omitted from this description the process of *frame repository update* upon encounter termination (whether after successful frame completion or due to frame selection failure). The previous frame repository \mathcal{F} has to be updated by computing new trajectory and substitution lists/counters for the respective (matching) frames or by adding a new frame if no existing frame trajectory matches the encounter just observed. Moreover, we apply *cluster validation* techniques as suggested in [4] to decide whether (and how) new observations should be merged into existing ones or whether they should be used to create a new frame. What remains to be specified is step 8, that is search strategy between different frames. Effectively, it is this search strategy that determines the degree of complexity reduction achieved by restricting the search space to a single active frame while looking for the next optimal action. One way to shape this search strategy is through long-term learning of the usefulness of particular frames.

4. Learning to Frame

For learning an optimal re-framing strategy, we use *hierarchical reinforcement learning* (RL) [2] techniques. In hierarchical RL, actions available in a general Markov Decision Process (MDP) are combined to yield macro-operators that can be applied over an extended number of decision

steps, the general idea being that compound time-extended policies, which (hopefully) optimally solve sub-problems of the original MDP, help to reduce the overall size of the state space. Using such macro-actions, an agent can use Semi-MDP (i.e. state history dependent) variants of learning methods such as Q-learning [14] to optimise its long-term ‘‘meta’’-strategy over these macro-policies.

An intuitive hierarchical RL approach that lends itself to an application to interaction frames particularly well is the *options* framework [8]. In this framework, an *option* is a triple $o = (\mathcal{I}, \pi, \beta)$ consisting of an input set $\mathcal{I} \subseteq \mathcal{S}$ of MDP states, a (stationary, stochastic) policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ over primitive actions \mathcal{A} and states \mathcal{S} , and a stochastic termination condition $\beta : \mathcal{S} \rightarrow [0, 1]$. 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 according to β . We can use this definition to re-interpret m¹nFFrA frames as options, where π is the (deterministic) strategy defined by determining m^* , and \mathcal{I} and β are defined by the validity, adequacy and desirability checks performed during the reasoning process of the previous section.

Let $s : \mathcal{M}_c^* \times 2^{\mathcal{L}} \rightarrow \mathcal{S}$ be some state abstraction function⁵ that returns a state for each pair (w, KB) of perceived encounter prefix w and belief KB . If we regard each frame $F \in \mathcal{F}$ as an option in the above sense, we can apply the Semi-MDP Q-learning update rule

$$\begin{aligned} Q(s, F) &\leftarrow (1 - \alpha)Q(s, F) + \\ &\alpha \left[\hat{R}(s, F) + \gamma \tau \max_{F' \in \mathcal{F}} Q(s', F') \right] \end{aligned} \quad (5)$$

where $s = s(w, KB)$ and $s' = s(ww', KB')$ are the states resulting from the encounter sequences w and ww' and the corresponding knowledge base contents KB and KB' as perceived between two re-framing decisions, α is an appropriately decreasing learning rate and τ is the number of steps during which F was the active frame (i.e. $\tau = |w'|$). Further, $\hat{R}(s, F)$ is the discounted reward accumulated in steps $t + 1, \dots, t + (\tau - 1)$.

Using the long-term utility estimates represented by Q , we can determine the optimal re-framing choice as

$$F^*(w, KB) = \arg \max_{F \in \mathcal{F}} Q(s(w, KB), F) \quad (6)$$

while applying a ‘‘greedy in the limit’’ infinite exploration strategy to avoid running into local minima. Note that learning re-framing Q-values in this way will allow for optimising framing decisions *within* encounters as well as *between* subsequent encounters, at least if there is some utility-relevant connection between them. Also, this re-framing strategy is not in itself sufficient to preclude infinite re-framing loops in the algorithm of section 3.2.⁶

⁵ It is unrealistic to assume that $\mathcal{M}_c^* \times 2^{\mathcal{L}}$ itself could be used as state space due to its unmanageable size, see also section 5.

5. Evaluation

The adequacy of our approach was tested in the Link Exchange Simulation system LIESON [1]. In this system, agents represent Web site owners who hold different views of the contents of other Web sites represented by numerical *private ratings*. At the same time, they can express their opinion about others' sites by attaching numerical weight labels to links laid toward these sites, so that these link weights function as *public ratings*. The physical actions available to an agent A are $addLink(A, B, R)$, $deleteLink(A, B)$ and $modifyRating(A, B, R')$ to add a link with public rating R to agent B , delete an existing link, or modify its current rating value to a new value R' .

The primary goal of agents in LIESON is to increase the dissemination of their own opinion through appropriate linkage structures, and for this purpose they negotiate with each other over mutually beneficial linkage.

The utility function in LIESON computes the popularity of each site on the grounds of a hypothetical model of Web user behaviour, according to which the probability of following a link is proportional to the numerical weight attached to a link. What is interesting about this utility function is that it yields very low utilities to all agents for empty, full negative and full positive linkage. This means that if agents do not lay any links at all, or if they lay links to every other site using uniformly maximal or uniformly minimal rating values for all links their performance will be very poor. On the other hand, if they truthfully link to every agent and display their true private rating of that site with every link ("honest linkage") or use a "politically correct" (PC) linkage scheme which is identical to honest linkage except that no links with negative rating values are laid, their performance is very high. Interestingly, PC linkage provides a substantially higher average utility than honest linkage, i.e. agents are better off concealing their discontent toward other sites.

LIESON agents reason about their actions along the following lines: Using their local link network knowledge, they project the usefulness of a number of physical actions and prioritise them using a goal/action queue in a BDI-like fashion. Then, they choose the topmost queue element for execution (unless its consequences have already been achieved or it is not applicable under current circumstances) and either (i) execute it themselves if this is possible or (ii) request its execution by an agent who can perform it (in the linkage scenario, this agent can always be uniquely identified). After such a request, the $m^2InFFrA$ component takes control of agent action until the initiated dialogue is termi-

nated and processes the frame repository and the perceived messages as described in the previous sections.

5.1. Proposal-based negotiation

In a first series of experiments, we ran simulations using a set of simple *proposal-based negotiation frames* that all agents were equipped with at the beginning and which allow for accepting and rejecting requests, making counter-proposals ("I can't do X as requested, but I can do Y for you instead") and reciprocally conditional proposals ("I will do X as you requested, if you do Y in return").

More precisely, frame repositories are initialised with frames that have the following trajectories:

$$\begin{aligned} &\langle \text{request}(A, B, X) \rightarrow \text{accept}(B, A, X) \rightarrow \text{confirm}(A, B, X) \\ &\quad \rightarrow \text{do}(B, X) \rangle \\ &\langle \text{request}(A, B, X) \rightarrow \text{propose}(B, A, Y) \rightarrow \text{accept}(A, B, Y) \\ &\quad \rightarrow \text{do}(B, Y) \rangle \\ &\langle \text{request}(A, B, X) \rightarrow \text{propose-also}(B, A, Y) \\ &\quad \rightarrow \text{accept}(A, B, Y) \rightarrow \text{do}(B, X) \\ &\quad \rightarrow \text{do}(A, Y) \rangle \\ &\langle \text{request}(A, B, X) \rightarrow \text{reject}(B, A, X) \rangle \\ &\langle \text{request}(A, B, X) \rightarrow \text{propose}(B, A, Y) \rightarrow \text{reject}(B, A, Y) \rangle \\ &\langle \text{request}(A, B, X) \rightarrow \text{propose-also}(B, A, Y) \\ &\quad \rightarrow \text{reject}(B, A, Y) \rangle \end{aligned}$$

Here, frame condition sets only require that actions X and Y can be physically executed wherever do-messages are involved; substitution lists are initially empty, and all trajectory occurrence and substitution counters are initialised to 0. The state abstraction function $s(w, KB)$ is defined using the *subject* of a conversation in terms of the link modification action(s) that are talked about (for details, please refer to [11]). Also, we use a very simple, strictly syntactical similarity measure σ that does not take any domain-specific similarities between actions or propositions into account [4]. Finally, the desirability threshold b (section 3.2) uses the *entropy-based heuristics* defined in [12] to trade off self-interested utility increase against conflict avoidance.

Selected results are shown in the plots of figure 1, in which the utility-per-round of the best/worst agent and the average utility of all agents is compared to the PC and honest linkage average agent utilities (shown as constant threshold lines labelled as "upper benchmark" and "lower benchmark"). The topmost plot illustrates what happens if non-adaptive (non- $m^2InFFrA$) agents take the fulfilment of the requests they issue for granted. These naively communicating BDI agents discover after a while that what others can do for them is better than what they can do themselves and cease all link modification activity. As in this sample run (the other plots are averaged over 100 runs each), all they do after a certain point in time is to issue requests toward others, and the system ends up in a deadlock state because no

6 This problem can be easily resolved by prohibiting repeated trial instantiation (validity/adequacy/desirability testing) of the same frame in a single re-framing iteration.

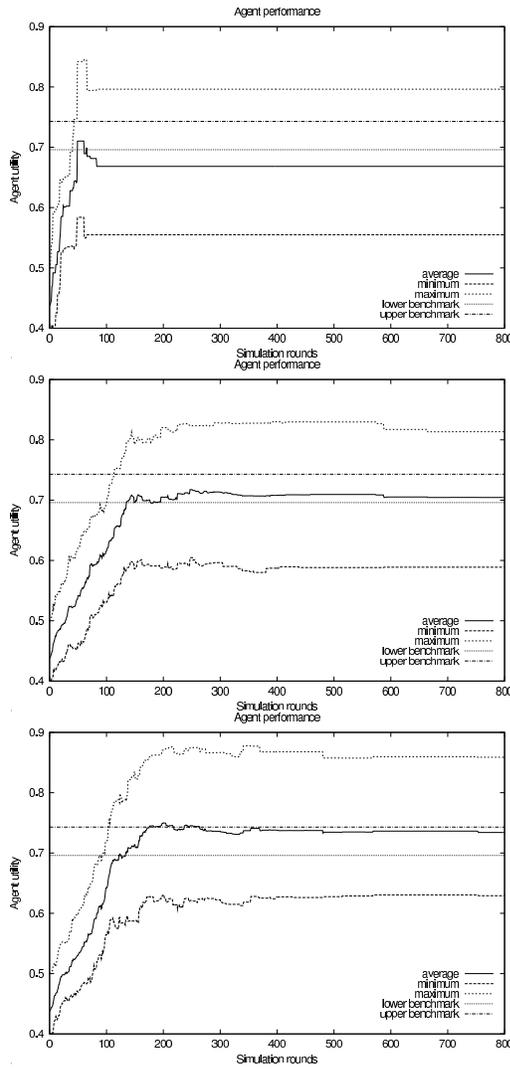


Figure 1. Performance plots: proposal-based negotiation. Top to bottom: naively communicating BDI agents, $m^2InFFrA$ agents without/with frame learning capabilities

agent is executing the actions others request him to. So, although these agents are rational in the sense that they give priority to those actions that seem more promising, they are bound to perform very poorly in open MAS in which adherence to pre-specified conversation patterns cannot be taken for granted.

In the two lower plots, we see that $m^2InFFrA$ agents succeed in doing better than the lower benchmark even if they are not endowed with frame learning capabilities, i.e. they choose a frame randomly when re-framing (middle plot). Added frame learning capabilities help attain the upper benchmark level on the average, and although this frame

learning is only an additional optimisation method, it does make a difference. These results are quite impressive considering that the agents have no prior knowledge about the effectiveness of the PC and honest linkage schemes. Also, they prove that $m^2InFFrA$ manages to integrate frame application and learning capabilities in a coherent social reasoning architecture that is able to operate successfully *in conjunction* with other (sub-social) agent activity in open MAS.

5.2. Interest-based negotiation

Despite the fact that the above results obtained with proposal-based negotiation frames are quite reassuring, these frames do not really qualify as complex knowledge-level conversation patterns that $m^2InFFrA$ is supposed to be useful for.

To investigate more complex interaction scenarios, we have therefore conducted a second series of simulations using *interest-based negotiation* (IBN) as proposed by [9] which is a specific approach to argumentation-based negotiation. In IBN, agents *challenge* other agents' proposals (in our case, requests or rejections of requests) by asking for reasons so that the other party is forced to *justify* her offer by proving that it does not concern unreachable or already achieved goals. The "challenger" party may in turn *attack* the justification by counter-arguments (suggesting wrong assumptions, threats to other goals, alternative means of achieving the same goal, etc.) in order to force their opponent to accept an alternative offer.

Based on IBN theory, Rovatsos [11] has designed two sets of frames for (i) single-shot argument exchange (i.e. negotiations that consist of just one challenge-justify-attack-concession iteration) and (ii) iterative IBN with several argument iterations. For lack of space, we shall only give a few examples for the single-shot case here.

Consider the following two frames (these are two of eleven frames defined for single-shot IBN in [11]):

$$\begin{aligned}
 & \langle \langle \overset{0}{\rightarrow} \text{request}(A, B, X) \overset{0}{\rightarrow} \text{reject}(B, A, X) \\
 & \quad \overset{0}{\rightarrow} \text{ask-reason}(A, B, \text{reject}(X)) \\
 & \quad \overset{0}{\rightarrow} \text{inf-threat}(B, A, T) \overset{0}{\rightarrow} \text{inf-alt}(A, B, Y) \\
 & \quad \overset{0}{\rightarrow} \text{concede}(B, A, Y) \overset{0}{\rightarrow} \text{request}(A, B, Y) \overset{0}{\rightarrow} \text{do}(B, Y) \rangle, \\
 & \langle \{ \text{can}(B, X), \text{goal}(B, T), \neg \text{achieves}(X, T), \text{achieves}(Y, T), \\
 & \quad Y \neq X, \text{can}(B, Y) \} \rangle, \langle \overset{0}{\rightarrow} \langle \rangle \rangle \rangle \\
 & \langle \langle \overset{0}{\rightarrow} \text{request}(A, B, X) \overset{0}{\rightarrow} \text{ask-reason}(B, A, \text{request}(X)) \\
 & \quad \overset{0}{\rightarrow} \text{inf-goal}(A, B, G) \overset{0}{\rightarrow} \text{att-goal}(B, A, \text{threat}(X, T)) \\
 & \quad \overset{0}{\rightarrow} \text{concede}(B, A, \text{threat}(X, T)) \rangle, \\
 & \langle \{ \text{can}(B, X), \text{goal}(A, G), \text{achieves}(X, G), \text{goal}(A, T), \\
 & \quad \neg \text{achieves}(Y, T) \} \rangle, \langle \overset{0}{\rightarrow} \langle \rangle \rangle \rangle
 \end{aligned}$$

In the first frame, B justifies her refusal to perform X by suggesting that X does not achieve T , whereupon A attacks

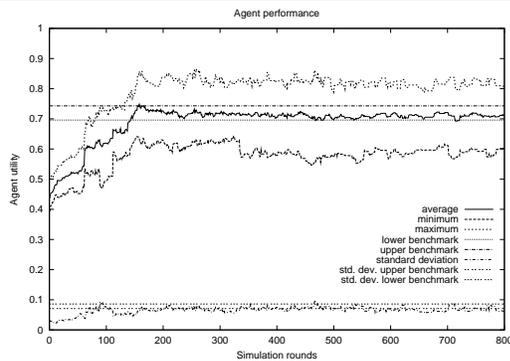


Figure 2. Performance plot: interest-based negotiation

by suggesting an alternative Y that does achieve T and B has to concede and perform Y . In the second, the requesting party A is asked to justify their request, but B attacks the alleged goal G by claiming that X at the same time does not achieve T (which is also a goal for A). A has to concede and no action follows.

Effectively, these IBN frames enforce a much stricter *communication regime* by forcing agents to justify their stance, to accept any alternative suggested for the same goal, to abandon any proposal that threatens at least one goal, etc. To make things even more difficult, we disallow re-framing for desirability reasons, i.e. agents no longer deviate from the provided frames for fear of low utility outcomes (but they still might abandon them if a failure occurs, e.g. when they fail to prove frame conditions that hold for the other party due to false/obsolete beliefs).

Figure 2 shows that the agents manage to maintain a reasonable level of long-term utility even under these circumstances (albeit with bigger fluctuations which indicate frequent “loss of an argument”), which illustrates that $m^2lnFFrA$ is capable of combining decision-theoretic learning with complex knowledge-based reasoning about constraint-governed conversation patterns. The results also suggest that IBN has an *equilibratory* effect on the social outcome since the utility difference between most and least successful agents is smaller than is the case in proposal-based negotiation. It deserves to be mentioned that, to our knowledge, the implementation of IBN frames in $m^2lnFFrA$ constitutes the first application of machine learning methods to argumentation-based negotiation in MASs, which underlines the flexibility and expressive power of our approach.

6. Conclusion

In this paper, we presented an integrated framework for reasoning about conversation patterns in multiagent systems that makes use of frame-based reasoning about inter-

action, hierarchical reinforcement learning of communication strategies and application as well as adaptation of communication pattern models using case-based reasoning and cluster validation techniques.

While this offers a new perspective for dealing with generic kinds of ACL and interaction protocol specifications in open systems (in which the designer cannot enforce obedience to a pre-specified semantics) by “making the best out of interaction experience”, a lot of work remains to be done. For example, we have to show that ACL/protocol semantics frameworks can be automatically converted to readily usable frame repositories. Also, we have to extend the expressiveness of frame models to cater for more expressive trajectory models that allow for branching and iteration (a first step in this direction has been taken in [11] by using online methods for concatenation of $m^2lnFFrA$ frames in a planning sense). Finally, “condition mining” techniques that allow for diagnostic reasoning in case of framing failure in order to adaptively modify the respective frame conditions will have to be investigated in the future.

References

- [1] LIESON homepage. <http://www7.in.tum.de/rovatsos/lieson>, 2002-2004.
- [2] A. Barto and S. Mahadevan. Recent Advances in Hierarchical Reinforcement Learning. *Discrete Event Systems*, 13:41–77, 2003.
- [3] F. Dignum, editor. *Advances in Agent Communication*, LNAI 2922, Springer-Verlag, 2004.
- [4] F. Fischer. Frame-Based Learning and Generalisation for Multiagent Communication. Diploma Thesis, Department of Informatics, Technical University of Munich, Germany, 2003.
- [5] F. Fischer and M. Rovatsos. Reasoning about Communication: A Practical Approach based on Empirical Semantics. *Procs. CIA-2004*, LNAI 3191, Springer-Verlag, 2004.
- [6] J. L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Francisco, 1993.
- [7] M. Kone, A. Shimazu, and T. Nakajima. The state of the art in agent communication languages. *Knowledge and Information Systems*, 2:259–284, 2000.
- [8] D. Precup. *Temporal Abstraction in Reinforcement Learning*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, 2000.
- [9] I. Rahwan, L. Sonenberg, and F. Dignum. Towards Interest-Based Negotiation. *Procs. AAMAS-03*, Melbourne, Australia, 2003.
- [10] A. S. Rao and M. Georgeff. An abstract architecture for rational agents. *Procs. KR&R-92*, 1992.
- [11] M. Rovatsos. *Computational Interaction Frames*. PhD thesis, Department of Informatics, Technical University of Munich, 2004.
- [12] M. Rovatsos, M. Nickles, and G. Weiß. Interaction is Meaning: A New Model for Communication in Open Systems. *Procs. AAMAS-03*, Melbourne, Australia, 2003.

- [13] M. Rovatsos, G. Weiß, and M. Wolf. An Approach to the Analysis and Design of Multiagent Systems based on Interaction Frames. *Procs. AAMAS-02*, Bologna, Italy, 2002.
- [14] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.