On the Organisation of Agent Experience: Scaling up Social Cognition[⋆]

Michael Rovatsos[1] and Kai Paetow[2]

[1] Department of Informatics, Technical University of Munich
85748 Garching bei München, Germany, rovatsos@cs.tum.edu
[2] Department of Technology Assessment, Technical University of Hamburg
21071 Hamburg, Germany, kai.paetow@tu-harburg.de

Abstract. This paper introduces "micro-scalability" as a novel design objective for social reasoning architectures operating in open multiagent systems. Micro-scalability is based on the idea that social reasoning algorithms should be devised in a way that allows for social complexity reduction, and that this can be achieved by operationalising principles of interactionist sociology. We first present a formal model of InFFrA agents called m^2InFFrA that utilises two cornerstones of micro-scalability, the principles of *social abstraction* and *transient social optimality*. Then, we exemplify the usefulness of these concepts by presenting experimental results with a novel opponent classification heuristic AdHoc that has been developed using the InFFrA social reasoning architecture. These results prove that micro-scalability deserves further investigation as a useful aspect of socionic research.

1 Introduction

The development of methods to improve the scalability of multiagent systems (MAS) is one of the central themes on the Socionics [11] research agenda. With the rapid growth of the Internet and of mobile communication technologies in recent years, large-scale *open systems* [5, 7] are becoming reality, and since scalability is a major concern in this kind of systems, it has the potential of becoming a key issue in MAS research in the next years. Roughly speaking, the various aspects of complexity in MAS that call for scalability fall into two categories [16]: (i) quantitative aspects that depend on the number of agents, interactions, etc. in a system and (ii) qualitative aspects such as the complexity of interactions and the heterogeneity of agents. According to the definition of Paetow, Schmitt and Malsch [16], scalability can be understood as the "operationalisation of complexity", i.e. the capacity of a system to manage complexity. Thus, a system scales well if it is capable of responding to increasing complexity appropriately, rather than "degrading ungracefully" or even collapsing. When complexity management is effective, it might even be possible to exploit increasing complexity rather than suffer from it.

A core theme in our own research has been to study the impact of using different social theories such as Luhmann's systems theory [10] and symbolic interactionism [2,

[⋆] This work is supported by DFG (German National Science Foundation) under contracts no. Br609/11-2 and MA759/4-2.

6, 12] as a “construction manual” for building different kinds of MAS with a particular focus on scalability properties of the resulting systems. While work on system-theoretic MAS architectures [14] can comment on arbitrary aspects of scalability, be it at the micro-, meso- or macro-level of social systems, multiagent research that seeks to exploit the principles of interactionism [15] has to focus on more specific aspects. Interactionist theories are, generally speaking, concerned with how social structures affect interaction between humans in a society, how individuals process social knowledge and how social sense is continually re-produced and potentially re-constructed through ongoing interaction. Interaction is seen as an exchange of symbols whose meaning is socially pre-determined, but needs to be re-affirmed by the actors who use these symbols to communicate. In a constant “struggle over signs”, it also undergoes changes. Quite naturally, as interactionism is concerned with how actors deal with social meaning, its focus lies not in the study of “mass phenomena”, the functioning of entire societies, or macro-characteristics of social life. So how can such a theory contribute to the construction of scalable MAS?

In this paper, we claim that MAS built using interactionist principles can provide substantial contributions to solving scalability problems through the concept of *micro-scalability*. Micro-scalability is the capacity of social reasoning methods to deal with heterogeneity and complexity in the social behaviour of other agents appropriately. Thus, micro-scalable agent architectures operationalise aspects of the complexity of interaction situations by using social cognition as an instrument to manage it. To achieve micro-scalability, we have to improve the ways social knowledge is processed at the cognitive level, and we discuss two principles that serve this purpose and that are both inspired by interactionism: the first one is *social abstraction*, a transition from the “opponent modelling” attitude (one of the main social reasoning perspectives traditionally assumed in MAS research) to a “modelling interaction situations” stance that abstracts from the mental properties of agents interacted with in favour of categorising situations and developing learning strategies to cope with these. The second is *transient social optimality*, which somewhat contradicts classical agent research principles of (constant) optimality. It means that in order to be comprehensible, socially intelligent agents have to abandon optimal strategies from time to time (in the long run, of course, their behaviour should still converge to optimal strategies). Both these methods were developed using interactionist theories, and they pave the way for further research on the subject.

The remainder of this paper is structured as follows: in section 2 we provide an overview of the social reasoning architecture InFFrA that serves as a general framework for developing micro-scalable agents. Section 3 introduces a formal model of InFFrA that is useful to formally describe social abstraction and transient social optimality. Then, in section 4, we present the opponent classification heuristic ADHOC based on InFFrA and discuss experimental results that prove the usefulness of the two principles empirically. Some conclusions are drawn in section 5.

2 The InFFrA social reasoning architecture

The Interaction Frames and Framing Architecture InFFrA is a framework for building social learning and reasoning architectures based on the notions of “interaction frames” and “framing”. Both are central concepts in the micro-sociological works of Erving Goffman [6], and InFFrA is an attempt to operationalise them for use with computational agents. Essentially, interaction frames describe classes of interaction situations and provide guidance to the agent about how to behave in a particular social context. Framing, on the other hand, signifies the process of applying frames in interaction situations appropriately. As Goffman puts it, framing is the process of answering the question *“what is going on here?”* in a given interaction situation – it enables the agent to act in a competent, routine fashion. Since the InFFrA framework has been described in detail elsewhere [15, 19, 21], we shall restrict ourselves to a fairly superficial description here.

2.1 Interaction Frames

In InFFrA, a frame is a data structure that contains information about

- possible courses of interaction (so-called *trajectories*) that are characteristic of the class of interactions described by the frame,
- *roles and relationships* between the parties involved in an interaction of this type (actors that fill certain roles, groups, representatives, etc.),
- *contexts* within which the interaction may take place (states of affairs before, during, and after an interaction is carried out) and
- *beliefs*, i.e. epistemic states of the interacting parties.

A graphical representation of an interaction frame is given in figure 1, with examples for possible contents of the four “slots” of information listed above. The “roles and relationships” slot contains, for example, graphical representations of groups (boxes) and relationships (arrows). The trajectory is represented by a protocol-like model of concurrent agent actions (in principle, though, trajectories may be given in any form of behavioural description). In the “context” slot, the trajectory model is “embedded” in boxes that contain preconditions, postconditions and maintenance conditions – these are propositions about properties of the environment that have to hold before, during, or after execution of the trajectory. A semantic network and a belief network are shown as two possible representations of ontological and causal knowledge in the “beliefs” slot, where shaded boxes define which parts of the networks are known to which participant of the frame.

It is characteristic of frames that certain attributes of the above must be assumed to be shared knowledge among interactants (so-called *common attributes*) for the frame to be carried out properly while others may be private knowledge of the agent who “owns” the frame (*private attributes*). Private attributes are mainly used by agents to store their personal experience with a frame, e.g. utilities associated with previous frame enactments and instance values for the variables used in generic representations that describe past enactments (“histories”), inter-frame relationships (“links”) etc.

An important thing to stress about the semantics of such a frame is that the trajectory slot constitutes its core element, because it describes *how* interactions following this frame will be carried out. All the other elements of the frame serve only as “side information” about conditions that will hold when this type of interaction occurs.

For example, a frame that describes wedding ceremonies contains information about the participating actors (bride, groom, best man, parents, guests, priest, etc.) and their

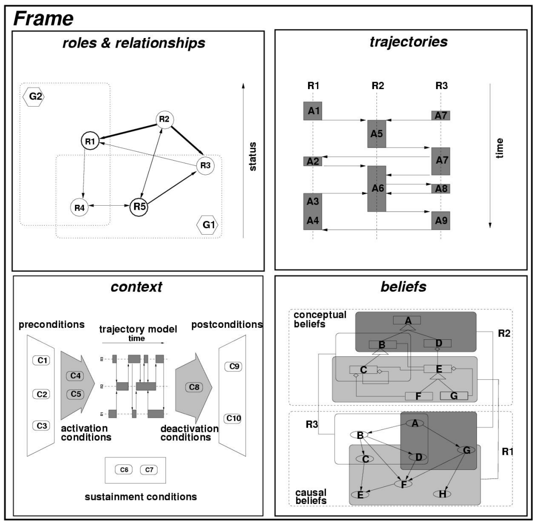

Fig. 1. An interaction frame.

relationships with each other (e.g. kinship between parents of bride and groom), conditions (pre-conditions: groom proposed to bride, invitations were sent out, etc. post-conditions: legal fact of “being married”, honeymoon; maintenance conditions: neither groom nor bride abandon the scene during the ceremony), and beliefs (love among bridal pair, agreement of parents, etc.). But although all this information is supplied, the ultimate use of the frame is to correctly interpret the actions that are observed in the ceremony, and – if in the position of one of the “active” participants – to act appropriately.

2.2 Framing

The second main element of InFFrA is the framing control flow model for social reasoning and social adaptation. It maintains interaction frames, modifies them with incoming observations if necessary, and applies the most suitable frame in a given interaction situation. In order to describe the steps performed in each framing cycle, some data structures need to be introduced. These are

- the *active frame*, the unique frame currently activated,
- the *perceived frame*, an interpretation of the currently observed state of affairs,
- the *difference model* that contains the differences between perceived frame and active frame,
- the *trial frame*, used when alternatives to the current frame are sought for,
- and the *frame repository*, in which the agent locally stores its frame knowledge.

Using these frame data structures, the framing component performs the following steps in each reasoning cycle during an interaction encounter:

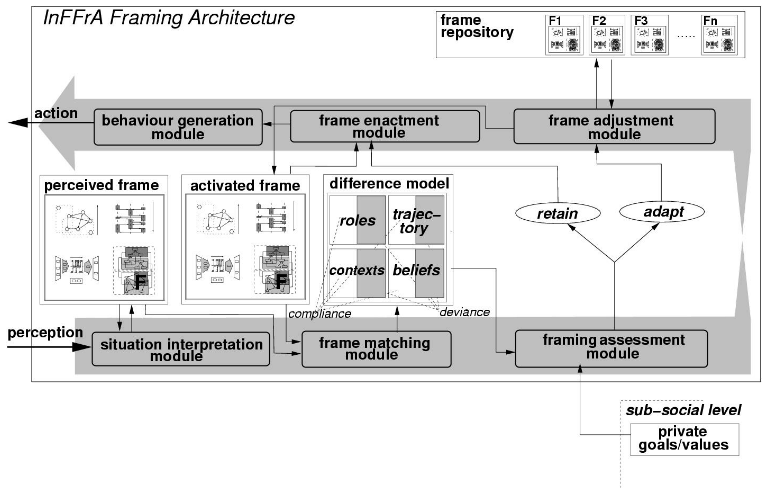

Fig. 2. An overview of the framing process in InFFrA with sub-processes and data structures.

1. *Interpretation & Matching:* Update the perceived frame, and compare it with the active frame (the normative picture of what the interaction should be like).
2. *Assessment:* Assess the usability of the current active frame in terms of
   (i) adequacy (compliance with the conditions of the active frame),
   (ii) validity (the degree to which the active frame trajectory matches the perceived encounter) and
   (iii) desirability (depending on whether the implications of the frame correspond to the agent's private goals).
3. *Framing decision:* If the active frame seems appropriate, continue with 6. Else, proceed with 4. to find a better frame.
4. *Re-framing:* Search the frame repository for better frames. "Mock-activate" them as trial frames iteratively and go back to 1; if no suitable frame is found, proceed with 5.
5. *Adaptation:* Iteratively modify frames in the frame repository and continue with 4.
6. *Enactment:* Influence action decisions by applying the active frame. Return to 1.

The entire framing process is visualised in figure 2. Apart from linking functional modules for each of the steps to the data structures on which these are performed, it connects the sub-social reasoning (e.g. BDI [18]) level to the InFFrA layer by taking agent's goals and preferences into consideration in the *frame assessment* phase.

InFFrA provides a unifying view for various perspectives on social learning at the interaction level, and has many features that allow to focus on the core aspects of developing social reasoning architectures. In particular, it assists the designer of such algorithms in focusing on the core issues, such as

– what representation to use for frames in a given application domain,
– how to store, retrieve and modify frames efficiently,

– how to intertwine local goal-directed reasoning with social commitment,
– which operators to provide for creative construction of frames by agents themselves,

and allows for integrating different outlooks on interaction, such as the machine learning perspective [21] and the agent communication semantics perspective [20].

2.3 Social Abstraction and Transient Social Optimality

Before describing the concepts of *social abstraction* and *transient social optimality* in terms of a formal model of InFFrA, we should informally explain the intuition behind them using the above elements of InFFrA.

The principle of *social abstraction* is fairly obvious in the above framework, since it is embodied by the very notion of interaction frames. By definition, these are thought to abstract from particular situations so as to capture the central distinctions between *classes* of these situations. More specifically, they abstract from particular interaction partners, specific world states, and may even coerce different actions in the trajectories into action types (e.g. by varying the content in speech acts with the same performative).

There are two principal arguments in favour of permitting such abstraction. Firstly, the argument from "pre-structuration" states that even though the possibilities for different interactions abound, there is only a certain number of relevant categories of interactions that occur over and over again. These are determined by the action and reasoning capabilities of the agents in a society, by the distribution of resources and by the available communication channels. By the second (much more practical) argument from "bounded rationality", agents have no other *choice* than to generalise from particular interactions for two reasons: (1) It is not reasonable to assume that they have arbitrarily elaborate reasoning capabilities to store all interaction experience and to consider all of that information to act optimally in a new encounter. (2) Even if this were the case, it seems unlikely that information could be directly re-used, given that (especially in open systems) encounters with the same interaction partners under the same circumstances are only occasional (in the best case – in the worst, they are one-time experiences).

As concerns *transient social optimality*, this point is somewhat harder to make. From the InFFrA architecture, it is obvious that optimal social decisions strongly rely on making the right assessments regarding the adequacy, validity and desirability of a candidate frame at the right time. However, in making the right framing choices, there are two conflicting goals that the agent needs to balance, namely (1) *predictability* and (2) optimal *utility*. On the one hand, an InFFrA agent wants to be able to predict others' imminent actions, and, on the other hand, it cannot stick to a particular predictable pattern of interaction if this is sub-optimal utility-wise.

In terms of the framing process outlined above, this conflict arises when adequacy, validity and desirability measures in frame assessment yield contradictory values. A standard way to proceed in that case would be to somehow weigh the importance of these measures so as to achieve an overall evaluation of candidate frames.

But since the agent's own framing choice also affects the reactions of other parties involved in the interaction, things are not that simple. If we assume that other agents are at least as socially intelligent as the agent in question, they will also record interaction experience and apply it strategically. So if we *deviate* from a given, established expectation (in the form of a "safe", well-known, stable frame), because its consequences are not desirable in the current state of affairs for the sake of "trying something new", it is very probable that we will not obtain predictable results. This is because peer interactants will be confused and unable to figure out how the interaction will turn out. So, even if – in the best case – others react in a way that is profitable for oneself, this will only happen at haphazard, i.e. it is not something the agent can rely on in decision making.

Transient social optimality is one answer to this problem that is based on neglecting promising alternatives occasionally for the sake of being "socially comprehensible" for others. In the framing process, this simply means that we trade desirability for validity and adequacy. Thus, the agent can hope to ensure predictability by sacrificing short-term utility, because it is better to have predictable opponents who maybe do not act as nicely as one would wish, rather than constantly trying to make optimal moves while the other might apply the same kind of strategic reasoning.

Goffman, in fact, stresses the strategic aspect of interaction, but it is taken to a level different from, e.g. the traditional decision-theoretic notion of "strategy optimisation" by assuming that agents *adopt* socially established procedures in a strategic fashion rather than to select particular actions in a utility-maximising fashion. To put it differently, behaviour during interactions is only rarely optimised by an individual by completely deviating from expectations; but choosing which of the different expectation patterns to activate is a highly strategic process in which agents compute optimal strategies before taking action.

In the formal model we will now present (section 3), we shall see that this can be formalised by avoiding "redundant" framing cycles unless major problems arise during an encounter. In section 4, the effectiveness of this approach will be underpinned by experimental results.

3 A Formal Model of InFFrA

m^2InFFrA is a full formal model of a specific kind of "simple" InFFrA agents that extends the model of "minimal" agents introduced in [20]. One of its most important aspects is that it enables us to formalise framing as a two-level Markov Decision Process (hence the m^2 in the name), through which the concept of transient social optimality can be defined more precisely. For lack of space, we will not present the model in full detail here (the interested reader should consult [4]) but focus on its core elements.

3.1 Basics

For starters, we assume the existence of two formal languages $\mathcal{L}$ and $\mathcal{M}$. $\mathcal{L}$ is a propositional logical language consisting of (i) atomic propositions $p, q(X, s), \ldots$ that may contain (implicitly universally quantified) variables and (ii) the usual connectives $\vee$, $\wedge$, $\rightarrow$ and $\neg$, the logical constants "true" and "false", and braces $()$ for grouping subexpressions together. Interpretations of formulae and entailment $\models$ in a *knowledge base* $KB \in 2^{\mathcal{L}}$ are defined in the usual way. We assume that m^2InFFrA agents maintain such a local knowledge base that is revised with incoming percepts and that they have sound and complete inference mechanisms for this logic at their disposal.

$\mathcal{M}$, on the other hand, is a language of message patterns (or templates). As in [20], messages observed in the system can be either physical messages ("real" actions) of the format $\mathtt{do}(a, ac)$ where a is the executing agent and ac is a symbol used for a physical action, or "non-physical" messages $performative(a, b, c)$ sent from a to b with content c. Both sender/recipient and content slots of messages may contain variables for agents, physical actions and content, but not for performatives. As we will soon show, this is useful to abstract from different observed messages. To discriminate between these patterns and actual messages, we write $\mathcal{M}_c$ for the language of *concrete*, variable-free messages.

As to the content c of a non-physical action, this can either be (i) an atomic proposition, (ii) a message term or physical action term, or (iii) a logical formula formed out of these elements. Effectively, this yields a variant of $\mathcal{L}$ as a content language that contains "propositions" for messages/actions (in the sense of "events"). Note that messages in $\mathcal{M}_c$ may, of course, still contain variables in the content slot that are variables in the sense of logical propositions. For the remainder of this paper, we will exclude this kind of logical variables from our considerations when talking about "variables" in the InFFrA sense. The interested reader can find full definitions of the languages $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{M}_c$ in [13] in this volume.

3.2 Interaction Frames

m^{2}InFFrA agents are agents that engage in discrete, turn-taking conversations between two parties, and maintain a frame repository $\mathcal{F} = \{F_1, \ldots, F_n\}$ in which they record knowledge about past interactions to apply this knowledge strategically in future encounters. These frames are defined as follows:

Definition 1. *A* frame *is a quadruple $F = (T, C, \Theta, h)$, where*

- *$T = \langle p_1, p_2, \ldots, p_n \rangle$ is the* trajectory *of the frame, a sequence of message terms (patterns) $p_i \in \mathcal{M}$;*
- *$\Theta = \langle \vartheta_1, \ldots \vartheta_m \rangle$ is an ordered list of substitutions $\vartheta_j = \langle [v_1/t_1], \ldots, [v_k/t_k] \rangle$ where each ϑ_j substitutes variables v_l by terms t_l;*
- *$C = \langle c_1, \ldots c_m \rangle$ is an ordered list of condition sets (sets of logical formulae) such that $c_j \in 2^{\mathcal{L}}$ is the condition set relevant under substitution ϑ_j;*
- *$h \in \mathbb{N}^{|T|}$ is an* occurrence counter *list counting the occurrence of each member of the trajectory T in previous encounters.*

The semantics of such a frame can be informally described as follows: the agent who "owns" F has experienced $h(p_1)$ encounters which started with a message matching the first element of the trajectory $p_1 = T(F)[1]$ (we write $T(F)$, $\Theta(F)$, $C(F)$ and $h(F)$ for functions that return the respective elements of F). $h(p_2)$ of these encounters continued with a message matching p_2, and so on. (This implies that there was no encounter with prefix $p_1 \cdots p_n$ that continued after p_n according to F.) We will sometimes use the abbreviated syntax $T_h(F) = \xrightarrow{h_1} p_1 \xrightarrow{h_2} p_2 \cdots \xrightarrow{h_n} p_n$ (where $h_n = h(p_n)$) to combine $T(F)$ and $h(F)$ in one expression.

Out of the h_n encounters that included the whole trajectory, exactly one substitution ϑ_j and one condition set c_j held true in the jth of these encounters. This means that $C(F)$ and $\Theta(F)$ capture the history of past encounters, in which the frame was executed as a whole; it also keeps track of "prefix-matching encounters" that ended after some initial portion of the trajectory, but does not maintain conditions and substitutions for these.

Note that the elements of frames introduced in Section 2 are present in this model, even if it has been simplified somewhat to admit formally rigorous treatment: Roles and relationships, context and beliefs are all captured in the condition sets in C. The trajectory is reduced to a simple sequence T of message templates; the history of the frame (and of its previous successful completions) is stored in C, Θ and h, and links between frames are implicitly maintained by cross-counting occurrence of prefixes of T.

Instead of going into the details of the formal semantics, an example shall illustrate what a frame means:

$$\begin{aligned} F = \Big\langle \big\langle &\xrightarrow{5} \texttt{propose}(A,B,X) \xrightarrow{3} \texttt{accept}(B,A,X) \xrightarrow{2} \texttt{do}(A,X)\big\rangle, \\ &\big\langle \{\mathit{self}(A), \mathit{other}(B), \mathit{can}(A,X)\}, \\ &\quad \{\mathit{agent}(A), \mathit{agent}(B), \mathit{action}(X)\}\big\rangle, \\ &\big\langle \langle [A/\texttt{agent\_1}], [B/\texttt{agent\_2}], [X/\texttt{pay\_price}]\rangle, \\ &\quad \langle [A/\texttt{agent\_3}], [B/\texttt{agent\_1}], [X/\texttt{deliver\_goods}]\rangle \big\rangle \Big\rangle \end{aligned}$$

According to this frame (we use the syntax $\langle T_h(F), C, \Theta\rangle$ instead of (T, C, Θ, h) for convenience), 5 encounters started with a message matching $\texttt{propose}(A, B, X)$, three of them continued with $\texttt{accept}(B, A, X)$ and two out of these were concluded by agent A performing physical action X. Thus, another two encounters may have terminated after the first message or were continued with a message that does not match $\texttt{accept}(B, A, X)$, as is the case for the encounter that turned out differently after the second message. Also, the agent has stored the two conditions and respective substitutions under which this frame has occurred ($h(T)[|h(T)|] = |C(T)| = |\Theta(T)|$).

3.3 Frame Semantics

The semantics of a frame in m^{2}InFFrA are given by the so-called *enactment constraint*, which is assumed to hold whenever a frame $F = (T, C, \Theta, h)$ exists. This constraint can be interpreted in two different ways: as a *retrospective* enactment constraint, that states how often certain transitions between messages have occurred in past encounters, and as a *prospective* enactment constraint, that provides an estimate for the probability with which an arbitrary message sequence is going to occur in the future.

Since the retrospective view is only used to make knowledge base inferences, we will restrict ourselves to a description of the prospective enactment constraint. Roughly speaking, it should express that we expect the future probability of a message sequence to be equal to the frequency with which it has been observed in the past. This could be achieved by simply computing transition frequencies h_{i+1}/h_i in all frames, and keeping track of the total number of encounters experienced so far. However, this would preclude any ability to generalise, since the probability of any message sequence never experienced before would be zero. Therefore, we introduce a real-valued *similarity* measure on message (pattern) sequences $\sigma : \mathcal{M}^* \times \mathcal{M}^* \rightarrow \mathbb{R}$ that adds a "case-based" flavour to frames and postulate that

$$\sigma(\vartheta, F) = \frac{1}{|\Theta(F)|} \sum_i \sigma(T(F)\vartheta, T(F)\Theta(F)[i]) \tag{1}$$

$$P(\vartheta|F) = \frac{\sigma(\vartheta, F)}{\sum_\chi \sigma(\chi, F)} \tag{2}$$

So the probability with which an arbitrary substitution ϑ is expected to occur if F is enacted is the expected similarity of ϑ determined using the past frequencies of the cases stored in F, normalised over all other possible substitutions χ. In other words, the more similar a substitution is to previous "samples" of F, the more likely it is to occur. This very much resembles the logic of case-based reasoning [8], because previous cases are combined and weighed according to their similarity with the current case in a way that resembles "nearest neighbour" heuristics.

Rather than knowing how probable ϑ is, we would like to know the probability of particular message sequences, if F is to provide any concrete guidance. For this purpose, we can compute

$$P(w) = \sum_{F \in \mathcal{F}, w = T(F)\vartheta} P(\vartheta|F)P(F) \tag{3}$$

where $P(F)$ is the posterior probability with which an encounter has matched an encounter.

With these definitions, we have formally defined a way to apply the principle of *social abstraction* when forming expectations about social behaviour. However, the above constraints only provide "observer semantics", and do not explain what m^{2}InFFrA agents who are actively involved in encounters should actually *do*.

3.4 Framing Agents

The following definitions provide the basis for describing the decision-making algorithm of a "framing" m^{2}InFFrA agent:

Definition 2. *An* agent *is a structure* $a = (\mathcal{L}, \mathcal{M}, \mathcal{E}, n, u, f, \kappa, \sigma)$ *where*

- $\mathcal{L}$, $\mathcal{M}$ *are the formal languages used for logical expressions and messages,*
- $\mathcal{E}$ *is a set of encounter identifiers,* $n \in \mathbb{N}$ *is the total number of encounters so far,*
- $u : 2^{\mathcal{L}} \times \mathcal{M}_c^* \rightarrow \mathbb{R}$ *is the agent's utility function estimate, where* $u(KB, w)$ *is the estimated utility of* w *being executed with knowledge base* KB*;*
- $f : \Phi \times \mathcal{M}_c^* \rightarrow \Phi$ *transforms any possible frame repository* $\mathcal{F} \in \Phi$ *to a new repository upon experience of an encounter* $e \in \mathcal{M}^*$ *(*Φ *is the set of all frame repositories);*
- $\kappa : 2^{\mathcal{L}} \times \mathcal{M}_c^* \rightarrow 2^{\mathcal{L}}$ *transforms knowledge base contents after an encounter;*
- *and* $\sigma : \mathcal{M}^* \times \mathcal{M}^* \rightarrow \mathbb{R}$ *is a* similarity measure *for variable substitutions.*

These rather complex definitions express that an agent is given by formal languages it uses for communication and reasoning and by utility estimates of communication (and action) sequences depending on his state of knowledge. Further, definition of an agent should specify how the agent transforms his frame repository and knowledge base upon experience of a new encounter, and what similarity function σ he uses to make predictions about future communications.

Given this agent design, the framing state $[a]$ of an agent who is currently experiencing an encounter starting with sequence w is a probability distribution over all potential consequences envisaged by a given his knowledge base and frame repository contents. Defining $[a]$ is, of course, a prerequisite for application of decision-theoretic principles (such as expected utility maximisation) in the design of m^{2}InFFrA agents. Also, the definition of $[a]$ embodies the empirical, constructivist and consequentialist view of communication semantics proclaimed in [20] in a single formula.

Definition 3. *Let* $a = (\mathcal{L}, \mathcal{M}, u, f, \kappa, \sigma)$ *an agent. A* framing state *of agent* a *is a function* $[a] : \Phi \times 2^{\mathcal{L}} \times \mathcal{M}_c^* \rightarrow \Delta(\mathcal{M}_c^*)$ *which maps every*

- *frame repository* $\mathcal{F} \in \Phi$*,*
- *current encounter prefix sequence* $w \in \mathcal{M}_c^*$*,*
- *current* knowledge base $KB \in 2^{\mathcal{L}}$

to a finite-support probability distribution $P \in \Delta(\mathcal{M}_c^*)$ *over future message sequences.*

To define $[a]$ in such a way that allows computation of an agent state in accordance with the semantics of m^{2}InFFrA frames, we exploit

- the fact that only frames whose trajectories match recently perceived messages need to be considered,
- current knowledge base contents to reduce the search space to those frames that are applicable in the current situation,
- information about substitutions already applied during the current encounter that restricts degrees of freedom in substituting variables,
- similarities of these substitutions to past cases stored in frames

to derive probabilities for future message sequences.

For lack of space, we cannot present the details of the derivation of $[a]$ here and have to refer the interested reader to [4]. However, we can briefly sketch some general ideas: If the sequence w has just been observed, this implies that under any $F \in \mathcal{F}$ that prefix-matches w some substitution has to be applied to perform this matching. For any such matching F, this will restrict "still possible" substitutions to a set $\Theta_{poss}(F, KB, w)$. These are all substitutions that provide values for the variables still free in the remaining steps of $T(F)$, under which

1. the remaining steps of $T(F)$ can still be (physically) executed,
2. there is at least one $c = C(F)[i]$ that can be satisfied by the contents of KB,

and which contain the smallest number of variables with this respect (this is necessary to implement a least commitment strategy).

With this, we can redefine equation 1 to obtain

$$P(\vartheta|F,w) = \frac{\sigma(\vartheta,F)}{\sum_{\chi\in\Theta_{poss}(F,KB,w)}\sigma(\chi,F)}$$

where

$$\vartheta \notin \Theta_{poss}(F,KB,w) \Rightarrow \sigma(\vartheta,F) = 0$$

such that the probability estimate is never positive if ϑ cannot be applied anymore.

Thus, after w,

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

yields the probability with which an encounter that started with w will be concluded with w'. As concerns estimates for $P(F|w)$, we can use the frequency with which the frame was carried out as a whole given all runs of F:

$$P(F|w) = \begin{cases} \frac{h(F)[|h(F)|]}{h(F)[|w|]} & \text{if } w \text{ can be unified with the first } |w| \text{ messages of } T(F) \\ 0 & \text{else} \end{cases} \tag{5}$$

Note that, if $w = \varepsilon$, the agent can also use this formula to check whether it is profitable to start an encounter, and what the best choice would be.

3.5 Transient Social Optimality in m^{2}InFFrA

The main advantage of m^{2}InFFrA is that it enables us to describe how frame-based design implements transient social optimality, and this is done by re-interpreting action selection in encounters as a Markov Decision Process (MDP) (see, e.g. [17]) and identifying two levels of decision-making in this MDP.

Formally, a (single-level) *Markov Decision problem* is given by a finite set of states S, a finite set of actions A, reward function $R : S \times A \rightarrow \mathbb{R}$ and a *transition probability function* $P \in \Delta(S \times A \times S)$. The intuition is as follows: in a sequence of stages, an agent observes the current state $s \in \mathcal{S}$, executes an action $a \in A$ and receives an immediate payoff $R(s,a)$. With probability $P(s'|s,a)$, the next state the agent finds himself in will be s'. A (so-called *stationary* and *stochastic*) *policy* is a mapping $\pi \in \Delta(S \times A)$ which specifies that the agent executes action a in state s with probability $\pi(s,a)$. The goal of an agent in an MDP is to maximise its long-term payoff, and a criterion that is often applied to define this goal is that of *infinite-horizon expected utility maximisation*, i.e. maximisation of the quantity

$$V^{\pi}(s) = E\left[\sum_{i=0}^{\infty}\gamma^i r_{t+i}\middle|\pi, s_t = s\right] \tag{6}$$

where $\gamma < 1$ is a discount factor a, $E[\cdot]$ denotes the expected value, and r_t is the reward achieved at the t-th step by applying π.

The MDP formalism has been very popular in recent years, and has to lead to the development of *reinforcement learning* [23] methods which provide algorithms for learning optimal policies from experience (i.e. executing actions and observing feedback values $R(s,a)$ and state transitions (s,a,s')). However, for realistic application domains it suffers from the problem of having to deal with huge state spaces S which leads to long convergence times for learning algorithms since every action has to be executed in every state (at least once) to be able to discern policies that are optimal.

Hierarchical reinforcement learning methods ([1] provides a survey of the state of the art), on the other hand, attempt to abstract from the state space of the "core" MDP in such a way that smaller state spaces are obtained at a higher level of abstraction.

With the concept of transient social optimality in mind, it only seems natural to view frames in a similar way, as "macro"-actions during the execution of which agents do not truly optimise overall their possible action choices in each step. So while equation 4 in the previous section can be basically decomposed to obtain a transition model similar to that of a ("flat") MDP, we can also use a frame as a *hierarchical abstraction of a reasonable course of interaction*. This view not only reduces decision-making complexity but also makes the decision-making agent comprehensible for its peers.

To develop this view, we have found the *options* framework [22] to be most suitable for application to m^{2}InFFrA. In short, the framework considers "options" $\langle \mathcal{I}, \pi, \beta \rangle$ where $\mathcal{I} \subseteq S$ is the so-called *initiation set*, π is the policy of the option, and $\beta : S^+ \rightarrow [0;1]$ is a stochastic termination condition. The idea is that an option is available at time t if and only if $s_t \in \mathcal{I}$. If it is chosen, then a_{t+1} is selected according to π, and $\beta(s_{t+1})$ determines whether execution of the option is terminated (whereupon the agent gets to choose a new option).

The concepts used to define options carry over to frames quite naturally:

- $\mathcal{I}$ is the set of states in which a frame can be used, which depends on its conditions and on whether it matches the current encounter prefix;
- π is given by the messages/actions the agent is supposed to execute according to the frame at specific points in time; of course, because of Θ_{poss}, each frame is not a single strategy, but a *set* of strategies, which are quite similar to each other;
- β is the criterion for "re-framing" – it depends on whether
  - the active frame still matches the perceived encounter,
  - the remaining active frame steps are still (physically) executable, and
  - on whether these remaining steps still appear desirable.

Looking at frames as options now enables us to apply standard reinforcement learning techniques such as Q-learning [24] to learn a *framing* strategy, while the freedom of choice at the level of substitutions from Θ_{poss} allows us to optimise during frame execution to find an optimal *action* strategy within the boundaries of a frame. Again, we are not able to go into the details of how this is achieved in practice. They can be found in [4].

To summarise, viewed from a decision-theoretic perspective, the concept of transient social optimality leads to a hierarchical view of decision-making and learning. Trying to stick to a "routine" so that others can understand what one is doing is, in other words, just a simplification of the process of making optimal decisions, since we

a_i \ a_j	C	D
C	(3,3)	(0,5)
D	(5,0)	(1,1)

Table 1. Prisoner's Dilemma payoff matrix. Matrix entries (u_i, u_j) contain the payoff values for agents a_i and a_j for a given combination of row/column action choices, respectively. C stands for each player's "cooperate" option, D stands for "defect".

deliberately disregard information which is available from experience for the sake of promoting stable patterns of interaction.

We believe it is one of the most interesting aspects of the work reported on here that the application of sociological concepts in the design of socially intelligent agents can be shown to parallel existing AI notions, such as "being hierarchical" about MDPs.

4 Micro-scalability in ADHOC

The ADaptive Heuristic for Opponent Classification ADHOC is an implementation of InFFrA that addresses the problem of learning opponent models in the presence of large numbers of opponents in game-theoretic interaction situations. It constitutes a first implementation of InFFrA in which we studied how agents can classify opponents they confront in fixed-length Iterated Prisoners' Dilemma ((I)PD) [9] games so as to learn optimal strategies against these opponent classes. If they succeed in classifying opponents quickly during the encounter, this would guarantee that they can behave optimally against unknown opponents.

A detailed description of ADHOC can be found in [21]. Here, we only provide an informal overview of the algorithms and concentrate on its relationship to m^2InFFrA and, in particular, on social abstraction and transient social optimality.

4.1 Overview

In the ADHOC interaction setting, agents from a growing population are randomly matched in pairs to play a fixed number (say, 10) of PD iterations. No agent knows the duration of each encounter, and, initially, all agents are unknown to each other. The goal of these agents is to maximise their cumulative utility over time, where one-shot payoffs are as in table 1. To this end, they evolve a (bounded) number of opponent classes from scratch that are the interaction frames of ADHOC. Each of these classes contains

1. A deterministic finite automaton (DFA) that represents the opponent's strategy in "strategic" mode, and the agent's own strategy in "comprehensible" mode.
2. A "support" of agents that belong to this class, i.e. that have played according to the DFA in past encounters.

3. A set of past encounters (“samples”) with members of this class that is used to train the DFA. To learn a DFA from sequences of actions, we apply the model-based learning method *US-L** proposed by Carmel and Markovitch [3].
4. A table of Q-values [24] that is updated using received payoffs in order to learn an optimal strategy against the DFA, as in *US-L**.

Also, ADHOC agents maintain a similarity function between agents and opponent classes that guides re-classification.

The ADHOC algorithm proceeds as follows: Given an opponent that the framing agent is currently interacting with, the behaviour of both agents in the current encounter (we assume that ADHOC is called after the encounter is over) and an upper bound on the maximal number of frames, the agent matches the current sequence of opponent moves with the behavioural models of the frames (situation interpretation and matching). It then determines the most appropriate class for the adversary (assessment) using the similarity function between adversaries and classes. After an encounter, the agent may have to revise its framing decision: If the current class does not cater for the current encounter, the class has to be modified (frame adaptation), or a better class has to be retrieved (re-framing). If no adequate alternative is found or frame adaptation seems inappropriate, a new class has to be generated that matches the current encounter. In order to determine its own next action, the agent applies the counter-strategy learned for this particular opponent model (behaviour generation). Feedback obtained from the encounter is used to update the hypothesis about the agent’s optimal strategy towards the current opponent class.

In “strategic” mode (the normal case), actions are selected in the following way: if an unknown peer is encountered, the agent determines the optimal class to be chosen after *each move* in the iterated game, possibly revising its choice over and over again in every step of the encounter. Else, the agent uses its experience with the peer by simply applying the counter-strategy suggested by the class this peer had previously been assigned to. This reflects the intuition that the agent puts much more effort into classification in case it interacts with a new adversary it knows very little about.

“Comprehensible” mode, on the other hand, is relevant when an agent discovers that the opponent is not pursuing any discernible strategy whatsoever, i.e. appears to be behaving randomly. This can be verified by checking whether the automaton of a class is constantly modified during many consecutive games. In this case, the ADHOC agent plays some fixed strategy for a fixed number of games, and then returns to strategic mode. So an agent takes the initiative to come up with a reasonable strategy, if his adversary’s behaviour makes no sense. In other words, he tries to become “learnable” himself in the hope that an adaptive opponent will develop some strategy that can be learned in turn.

4.2 ADHOC and $\mathsf{m}^2\mathsf{InFFrA}$

Although ADHOC complies with the formal model of $\mathsf{m}^2\mathsf{InFFrA}$ for the most part, some aspects of InFFrA have been realised in it at a higher level of complexity (marked with $\oplus$ below) and some in a much simpler fashion (marked with $\ominus$). These deviations from the formal model are due to the properties of the interaction scenario:

- $\ominus$ $\mathcal{M}_c$ is reduced to two actions C(ooperate) and D(efect), since there are no more actions in the (I)PD game. Both of these messages are also physical actions that always yield payoffs to the players.
- $\oplus$ Trajectories are modelled as deterministic finite automata (DFA) rather than simple trajectories. This way, trajectory models are much more expressive[1], and this is computationally feasible because there are only two actions. Note, however, that these DFA only define the strategy of *one* party while the other party is *free* to behave in arbitrary ways without breaking a frame.
- $\ominus$ Condition sets in frames are only used to discriminate between "strategic" and "comprehensible" mode, i.e. any frame can only be activated in one of these modes. Otherwise, there are no restrictions as to when an opponent class/frame might be assigned to an opponent.
- $\ominus$ Substitutions only refer to agent names (since messages C and D cannot be parametrised), and all substitutions of a frame/opponent class are identical. Consequently, the similarity measure σ can be re-defined to directly compare an opponent to an entire opponent class.
- $\oplus$ Private attributes are more elaborate than in standard $\mathsf{m}^2\mathsf{InFFrA}$. Apart from a fixed number of encounter samples that agents store with each frame, they also maintain a membership function that assigns each agent to a class. Moreover, all agents' similarity with all classes it constantly tracked. Utility experience is stored in a Q-table, which is also used to guide action in strategic mode.
- $\oplus$ ADHOC agents generate frames (opponent classes) from scratch, and the architecture not only defines how frames are modified with experience, but also when new frames should be created, merged or deleted.
- $\oplus$ In comprehensible mode, the agent plays the fixed strategy represented by the DFA of the frame activated. In strategic mode, however, the agent is free to optimise its behaviour, and selects actions according to the exploration/exploitation factors of the Q-table that belongs to the class the opponent has been assigned to.

Placing ADHOC in the $\mathsf{m}^2\mathsf{InFFrA}$ context in this way allows for a more precise identification of the social abstraction and transient social optimality properties of ADHOC.

Social Abstraction. This is achieved by viewing frames (opponent classes) as possible social behaviours that do not depend on the particular agent who employs them. Experimental results prove that this is not only a reasonable strategy in the light of bounded rationality that allows agents to maintain a bounded number of opponent models although they are faced with huge numbers of opponents. Much more than this, coercing different opponents into the same class even leads to an *accelerated* learning process. As shown in figure 3, ADHOC agents converge much more quickly to high payoffs than agents who maintain one model per opponent.

This is yet another advantage of social abstraction: by categorising interaction situations appropriately, an agent can learn optimal strategies for his own behaviour much

[1] Note that, since encounters have a fixed length, the DFA-frame could be replaced by a number of frames with sequence trajectories that are semantically linked with each other to express that they pertain to the same strategy.

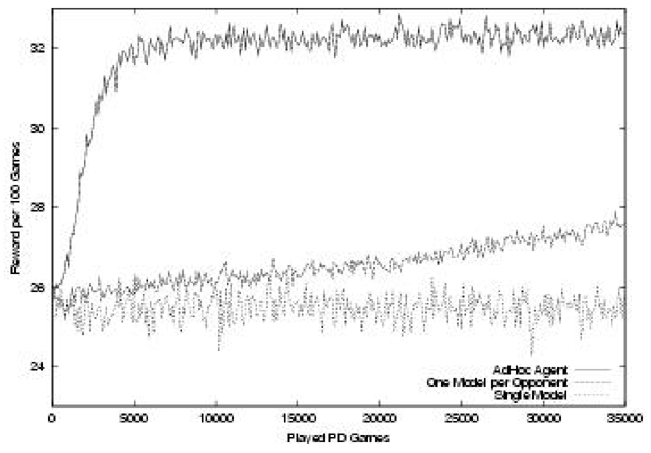

Fig. 3. Comparison of cumulative rewards between an ADHOC agent, an agent that maintains one model for each opponent and an agent that has only a single model for all opponents.

easier, because more "data-per-model" is available (in ADHOC, this means that the Q-tables are updated much more often, and thus converge to optimal strategies more quickly). This plot also shows, however, that it is not possible to combine all the behaviours into just one opponent model, as long as more than one strategy is around. Figure 4 provides further evidence for the facts that ADHOC learns exactly as many classes as are present in the long term. Here, ADHOC agents are shown to converge to four opponent classes in a setting where opponent agents use any one of four fixed strategies (ALL C, ALL D, TIT FOR TAT or TIT FOR TWO TATS). In terms of scalability, this is a very important result, because it means that ADHOC agents are capable of evolving a suitable set of opponent classes regardless of the size of the agent population, as long as the number of strategies employed by adversaries is limited (and in most applications, this will be reasonable to assume). A particular challenge in developing the heuristic in a way that guarantees this convergence is, of course, to ensure that similar classes are merged in the long run, so that unnecessary "temporary" classes can be erased (these are generated when encountering new agents or when the DFA-learning algorithm makes a wrong guess).

Transient Social Optimality. This aspect of micro-scalability appears in ADHOC in a twofold way. Firstly, it is embodied in the strategy of "blindly" selecting a frame that an agent has been assigned to, if that agent is encountered again (which is followed until that opponent deviates from his previous strategy). Assuming that the ADHOC agent has been using this frame for a while, he will learn to play an optimal strategy against it, and his own behaviour towards this class of opponents will be stable. Thus, implicitly, he can "inform" his opponents about the strategy he will settle on if they keep behaving the same way, so that he becomes more predictable for these agents in turn. If an unknown agent is encountered, this strategy (e.g. picking an arbitrary frame)

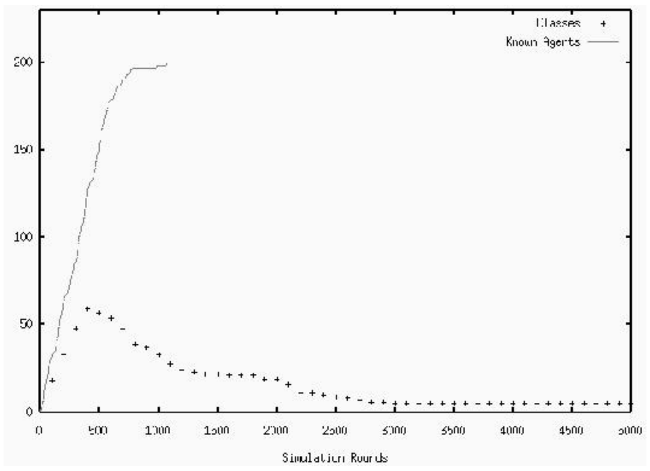

Fig. 4. Number of agent classes an ADHOC agent creates over time in contrast to the total number of known (fixed-strategy) opponents (which is increased by 40 in rounds 150, 300 and 450). As can be seen, the number of identified classes converges to the actual (four) strategies.

would be too risky, so transient optimality cannot be applied in this case, and the agent must select the best-matching class in each round to make an optimal move.

In terms of m^2InFFrA, this means that the criterion of whether to activate the frame with the highest expected utility is "in each move" if the opponent is unknown and "only at the beginning of the encounter" if the agent is acquainted with the current adversary.

But there is a second, much more important aspect of transient optimality in ADHOC, and this is the process of switching between strategic and comprehensible modes. This process has been elaborated after initial experiments with ADHOC agents playing agents each other (rather than playing against simple agents with fixed strategies as above), where agents appeared to behave randomly throughout. This was due to the fact that, if they want to learn optimal strategies, this will involve some form of exploration. Unfortunately, whenever agents perform exploratory actions their behaviour can no more be represented as a DFA, and can therefore not be classified by their opponents.

To alleviate this problem, agents were made to switch to a fixed strategy for a while, whenever they cannot understand what their opponent is doing. As the results in figure 5 show, this was sufficient to achieve effective patterns of interaction in the "ADHOC vs. ADHOC" case. So, by abandoning optimality as a foremost goal in certain situations, an ADHOC agent becomes comprehensible, his opponents settle on a counter-strategy, and so does the first agent after he switches back to strategic mode.

Returning to the m^2InFFrA model, we can see that frame activation does not depend on (payoff) optimality *at all* in this case. Instead, some heuristic is used to determine the activated frame, and this frame remains activated without making a new framing decision for a while. A final interesting property of these simulations is that, although TIT FOR TAT outperformed other strategies as a choice for the comprehensible strategy

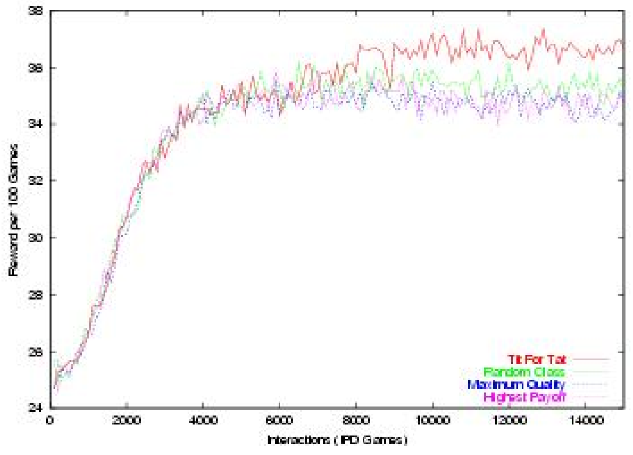

Fig. 5. Comparison of agent performance in "AdHoc vs. AdHoc" simulations and different selection methods for the strategy chosen if the opponent exhibits random behaviour. TIT FOR TAT can be shown to perform slightly better when chosen as an interim fixed strategy, while other heuristics were based on generating random DFA, choosing the DFA from the opponent class with maximum payoff, or that with highest "quality" (a heuristic function).

(which is not surprising because TIT FOR TAT is very powerful against a variety of counter-strategies), the result of achieving fruitful coordination does not depend on the interim fixed strategy choice. This suggests that being comprehensible at all is much more important for framing agents than the short-term payoffs ensured by following an effective comprehensible strategy.

5 Conclusions

This paper introduced "micro-scalability" as a new concept for social reasoning architectures based on the principles of *social abstraction* and *transient social optimality*. Micro-scalability constitutes the central contribution of MAS architectures derived from interactionist social theories to the scalability aspect of the Socionics endeavour. Starting from the social reasoning architecture InFFrA that is inspired by Goffmanian concepts, we provided an informal description of these notions. Then, they were made precise by introducing a formal model of InFFrA, and set into the context of simulation experiments obtained in the development of the opponent classification heuristic AdHoc.

Eventually, we hope that learning algorithms that are currently being developed using the m^2InFFrA model will prove scalable in more realistic applications, and we are currently working toward this goal. Other interesting implications are the evolution of stable empirical agent communication semantics using frame-learning, and a deeper investigation into the macro-effects of using micro-scalable social reasoning architectures.

References

1. Andrew Barto and Sridhar Mahadevan. Recent Advances in Hierarchical Reinforcement Learning. *Discrete Event Systems*, 13:41–77, 2003.
2. H. Blumer. Society as Symbolic Interaction. In A. M. Rose, editor, *Human Behavior and Social Process*. Routledge and Kegan Paul, London, 1962.
3. D. Carmel and S. Markovitch. Learning and using opponent models in adversary search. Technical Report 9609, Technion, 1996.
4. F. Fischer. Frame-Based Learning and Generalisation for Multiagent Communication. Diploma Thesis, Department of Informatics, Technical University of Munich, Munich, Germany, December 2003.
5. L. Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47:107–138, 1991.
6. E. Goffman. *Frame Analysis: An Essay on the Organisation of Experience*. Harper and Row, New York, NY, 1974. Reprinted 1990 by Northeastern University Press.
7. C. Hewitt. Open information sytems semantics for distributed artificial intelligence. *Artificial Intelligence*, 47:79–106, 1991.
8. J. L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Francisco, 1993.
9. R. D. Luce and H. Raiffa. *Games and Decisions*. John Wiley & Sons, New York, NY, 1957.
10. N. Luhmann. *Social Systems*. Stanford University Press, Palo Alto, CA, 1995.
11. Th. Malsch. Naming the Unnamable: Socionics or the Sociological Turn of/to Distributed Artificial Intelligence. *Autonomous Agents and Multi-Agent Systems*, 4(3):155–186, 2001.
12. G. H. Mead. *Mind, Self, and Society*. University of Chicago Press, Chicago, IL, 1934.
13. M. Nickles and M. Rovatsos. Communication Systems: A Unified Model of Socially Intelligent Systems. In this volume.
14. Matthias Nickles and Kai F. Lorentzen. Multiagent Systems without Agents – Mirror-Holons for the Derivation and Enactment of Functional Communication Structures. In this volume.
15. K. Paetow and M. Rovatsos. Grundlagen einer interaktionistischen Sozionik ("Foundations of Interactionist Socionics"). Research Report RR-8, Department of Technology Assessment, Technical University of Hamburg, Hamburg, February 2003.
16. Kai Paetow, Marco Schmitt, and Thomas Malsch. Scalability, scaling processes and the management of complexity. a system theoretical approach. In this volume.
17. M. L. Puterman. *Markov Decision Problems*. John Wiley & Sons, New York, NY, 1994.
18. A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 312–319, 1995.
19. M. Rovatsos. Interaction frames for artificial agents. Technical Report Research Report FKI-244-01, Department of Informatics, Technical University of Munich, 2001.
20. M. Rovatsos, M. Nickles, and G. Weiß. Interaction is Meaning: A New Model for Communication in Open Systems. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, Australia, 2003.
21. M. Rovatsos, G. Weiß, and M. Wolf. Multiagent Learning for Open Systems: A Study in Opponent Classification. In E. Alonso, D. Kazakov, and D. Kudenko, editors, *Adaptive Agents and Multi-Agent Systems*, volume 2636 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, 2003.
22. R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112:181–211, 1999.
23. R.S. Sutton and A.G. Barto. *Reinforcement Learning. An Introduction*. The MIT Press/A Bradford Book, Cambridge, MA, 1998.
24. C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.