An Approach to the Analysis and Design of Multiagent Systems Based on Interaction Frames

Michael Rovatsos, Gerhard Weiss, Marco Wolf
{rovatsos, weissg, wolf}@cs.tum.edu
Faculty of Informatics
Technical University of Munich
80290 Munich, Germany

ABSTRACT

This paper introduces InFFrA, a novel method for the analysis and design of multiagent systems that is based on the notions of *interaction frames* and *framing*. We lay out a conceptual framework for viewing multiagent systems (MAS) as societies consisting of socially intelligent agents that record and organise their interaction experience so as to use it strategically in future interactions. We also provide criteria for the class of MAS InFFrA is suited for. The benefits of our approach are that it helps to understand and develop socially intelligent agents as well as to identify shortcomings of existing MAS. The method is evaluated through the analysis of an opponent classification heuristic that is used to optimise strategic behaviour in multiagent games, and interesting issues for future research are discussed.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; D.2.1 [**Software Engineering**]: Requirements/Specification—*Methodologies*

General Terms

Design, Theory

Keywords

Social reasoning architectures, theories of agency and autonomy, sociologically grounded methods, social order, control & norms, coordinating multiple agents & activities.

1. INTRODUCTION

Multiagent systems (MAS) research has long viewed societies of interacting agents as either relatively closed, well-organised compounds consisting of subordinate entities that work towards a common goal or as open systems populated by self-interested agents which maintain "arms-length relationships" with each other while pursuing their own (potentially conflicting) goals.

In this paper, we suggest a novel outlook on this dichotomy that is located at the very borderline between cognitive intra-agent reasoning and societal, supra-agent interaction processes. Inspired by Erving Goffman's micro-sociological analyses of everyday interaction processes in human societies [5], we propose a socio-centric view of interaction processes that is based on the notions of *frames* and *framing*. Frames are units of knowledge that reflect the regularities in interaction processes with respect to recurring patterns of action among interacting actors. While they are not directly subject to manipulation by individual actors (since their validity depends on others' behaviour), the importance of cognitive processing of such social knowledge comes in when agents engage in *framing*, i.e. in the activity of acquiring and adapting frame knowledge through interaction experience and strategically applying this knowledge in order to achieve their own private goals.

Thus, instead of focusing on either the closed or open view of MAS, we propose a view whose foremost aim is to focus on the necessity for agents to cope with the constraints interaction practices impose on their individual behaviour, or, in other words, on societies of completely *autonomous* yet *socially bounded* agents.

We have developed a model of computational *interaction frames* [8] that is capable of adequately capturing the knowledge that is necessary for agents to understand "what is going on" in their encounters with others, and a conceptual architecture called InFFrA (**In**teraction **F**rames and **Fr**aming **A**rchitecture) that re-formulates the concept of *framing* in computational terms. Our hypothesis is that this architecture is sufficient to model and analyse a very general class of social reasoning algorithms, both at the cognitive and at the social level. Its power lies in providing a clear and intuitive conceptual framework (i) for decomposing social reasoning algorithms in terms of a generic view of framing, and (ii) for describing global MAS behaviour in terms of interleaving framing processes. It thus provides a basis for analysing and improving social reasoning methods, and also has the potential for developing new kinds of socially intelligent autonomous agents.

The remainder of the paper is structured as follows. Section 2 describes the class of MAS our approach is suited for. Section 3 presents a computational model of frames, and Section 4 introduces the framing agent architecture InFFrA.

AAMAS'02, July 15-19, 2002, Bologna, Italy.

This is followed by an application of our method to an opponent classification system in the context of multiagent games in Section 5. Section 6 rounds up with some conclusions and an outlook on directions for further research on the subject.

2. FRAMING MULTIAGENT SYSTEMS

Before introducing computational interaction frames and a framing architecture, we should first describe what kind of agent-based social reasoning systems they are suited for.

Generally, we assume that MAS consist of self-interested agents that can neither predict the behaviour of their partners/adversaries, nor rely on their benevolence or on their capabilities. This means that they are confronted with co-actors whose behaviour is *contingent*, and the only way to overcome this problem is by using *communication*. Thereby, any action qualifies as *communicative* that (i) is explicitly performed by an actor, (ii) is observable by at least two parties and (iii) *signifies* or *stands for* other actions by virtue of having been repeatedly followed by those other actions in the past (actually, it stands for the *expectation* [1] that they will be performed).

More specifically, we require that agents in the MAS we examine *reason about* patterns of communication experienced in social interaction and *use* them strategically when pursuing their own goals. So the minimal cognitive-level requirement for "framing" MAS is that they contain (at least some) agents that can

1. record communication and identify patterns and regularities within it,
2. relate the effects of communicative behaviour to one's private goals by assessing its consequences,
3. engage in goal-oriented communication with others by exploiting behavioural regularities among agents.

We claim that InFFrA is capable of adequately modelling this specific aspect – the *socio-cognitive interface*, so to speak – of MAS that exhibit these features. The patterns and regularities in communication are captured by the notion of computational interaction frames, and strategic usage of these frames is captured by the process of framing.

3. INTERACTION FRAMES

Frames [5] are a central concept in the micro-sociological analyses of everyday life that Erving Goffman carried out; they can be seen as the answer to the simple question *"what is going on here?"* that each human poses to herself in any interaction situation. That is, they provide "framing" information about a particular class of interaction situations that will allow the participant to act appropriately, i.e. in a competent, routine fashion.

In a MAS context, we can view them as data structures that contain sufficient knowledge to structure interaction for the individual that employs them. In that, they describe what is *in-between* rather than what is *inside* agents, as most mentalistic approaches do. While offering this advantage of being genuinely social, they are still linked to the mental processes of the agent which is using them, and hence allow for the modelling of rational, self-interested agents.

What kinds of information will such a data structure consist of? If a frame is to feed the interaction with sufficient information, it must exhibit the following properties:

1. *Common knowledge:* It must allegedly be *shared* knowledge among the interacting agents. When one agent uses it, it must assume its peer(s) to have the same information.
2. *Relevance:* The interaction knowledge captured by the frame must be grounded in agents' experience, it must occur repeatedly, and it must affect the agents' standing.
3. *Generalisation:* It must generalise from particular enactments of a class of interactions.
4. *Instrumentalisation:* The knowledge captured by it must relate to the agent's private goals and preferences, if the agent is to gain from using frame knowledge.

Requirements 2 and 3 refer to qualitative aspects of the knowledge a frame captures. They require that agents identify what matters to them through their experience of past encounters, and that they be able to store those pieces of information efficiently in expressive representations.

Requirements 1 and 4 can be realised by separating frame knowledge into *common* and *private* attributes, the former employed in reasoning "as if" interaction partners had the same knowledge, the latter expressing the agent's current stance towards the frame in question.

3.1 Common attributes

In principle, it would be sufficient for a "framing" agent to record sequences of experienced communication, and, in fact, this approach has been taken[1] in [1]. However, we observe that it is both counter-intuitive and inefficient for an agent to store huge numbers of interaction instances, and that there is a need for deriving more compact representations by generalising from particular instances.

The five attributes we have derived serve precisely this purpose: *roles and relationships* parametrise interaction sequences with knowledge about the participating parties, *contexts* generalise over the state of affairs before, during, and after an interaction is carried out, and *beliefs* generalise over epistemic states of the involved actors. Together with the core interaction *trajectory* ("what actually happens"), these three attributes form an interaction frame. *Repositories* of such frames are organised by relating frames to each other by virtue of meta-frame *links*.

In the following paragraphs, these classes of common attributes are described in more detail.

3.1.1 Trajectories

Trajectories are the core element of interaction frames, as they describe what the frame is *about*. In essence, they are temporally ordered communicative action sequences of actors that relate to each other, as laid out in Section 2. The most genuine kind of such trajectories are, of course, messages in the context of communication protocols, but also any other kind of communicative action in the sense described above. We use generalised *trajectory models* with variables for actors and for the parameters of actions that occur in them (rather than overtly specific traces of interactions), in order to cover a suitably wide range of situations.

[1]Though not from an agent perspective (there, it was used to model the evolution of an entire MAS from the perspective of an external observer).

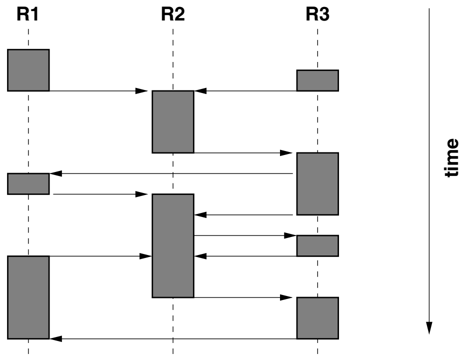

Figure 1: A trajectory model.

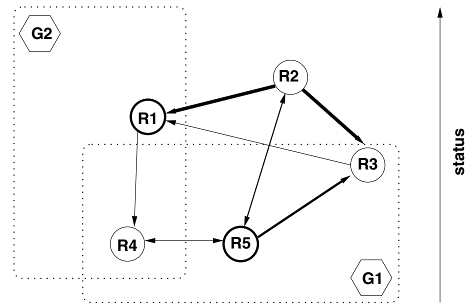

Figure 2: Role and relationship model.

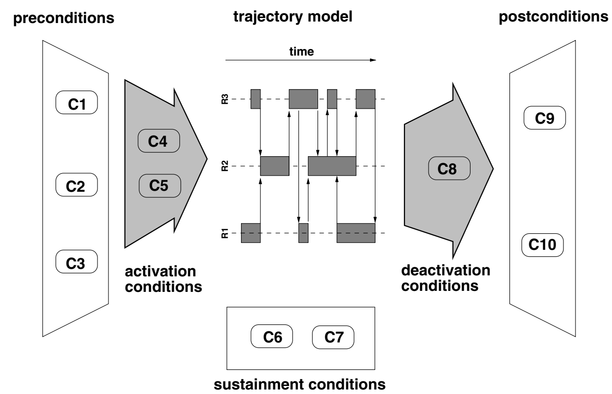

Figure 3: Context model with embedded trajectory that shows conditions C_i in shaded (relevance) and white (enactment) boxes/arrows.

Instead of using a specific formalism for expressing such trajectories at this point we have chosen to employ merely abstract graphical models[2] of protocol-like trajectories as the one depicted in Figure 1: in it, three actors (actually roles) R_1, R_2, R_3 perceive each other's actions and react upon them. Arrows denote messages, shaded boxes observable non-message ("physical") actions.

3.1.2 Roles & Relationships

Like trajectories generalise from individual courses of action, roles abstract from individuals by describing properties of classes of actors. We suggest that a powerful concept of *role models* as data structures should encompass three kinds of attributes that define a role R:

1. *Behavioural attributes*: expected behaviour, skills/capabilities/access to resources;
2. *Intentional attributes*: beliefs, desires/preferences, intentions; goals, tasks;
3. *Social attributes (Relationships)*: dependencies/power (on/over others), aggregation (groups/organisations) and membership, representation (acting on behalf), acquaintance (knowledge about others).

The graphical model we use here (cf. Figure 2) shows agent roles as rounded nodes and group roles as hexagons with a boundary box around members (possibly overlapping); these are interlinked through relationship arcs (for various types of relationships). The vertical line may be used as a status scale, if a one-dimensional measure has been defined (e.g. by computing the total of existing dependencies for each role).

3.1.3 Contexts

Having abstracted from actions and actors, there is also a need to abstract from *situations* in which the interactions occur, and this is achieved by using *context models*. These consist of two parts:

1. Activation and deactivation conditions: these are *relevance* conditions, i.e. conditions under which the frame will be adopted or abandoned by the participating parties.
2. Pre-, post- and sustainment conditions: these are *enactment* conditions, conditions that have to hold *before* the frame can be carried out, conditions that are always ensured *after* a frame has been completed, and conditions that must hold *throughout* the enactment of a frame.

As Figure 3 suggests, the scope of a trajectory is defined by embedding it into a context model: Relevance conditions define clear-cut relevance conditions for adoption and abandonment of the frame; enactment conditions supply information about what is needed to carry out the frame properly and about what the frame achieves.

3.1.4 Beliefs

According to our theoretical intuitions, the *beliefs model* plays a subordinate role in a frame. Although it may contain beliefs that are necessary to execute and interpret the frame properly, e.g. causal or conceptual knowledge (as Figure 4 suggests), it can be neglected as long as the inter*action* itself occurs as expected. However, if an actor is able to infer certain beliefs that his peers have when enacting a particular frame, or if it is able to associate certain beliefs of his own with the frame in a useful way, beliefs may aid the frame-wise organisation of experience.

3.1.5 Links and history.

Links that relate an entire frame to other frames by relationships such as aggregation, inheritance (in the object-

[2]Since the adequacy of specific formalisms may very across applications, this strategy is followed throughout the paper.

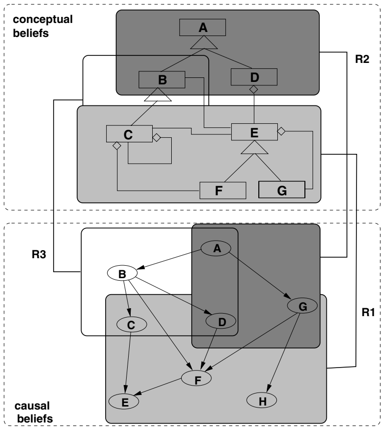

Figure 4: Two-part belief model with conceptual and causal beliefs. Roles' beliefs are depicted as shaded sub-areas of the networks.

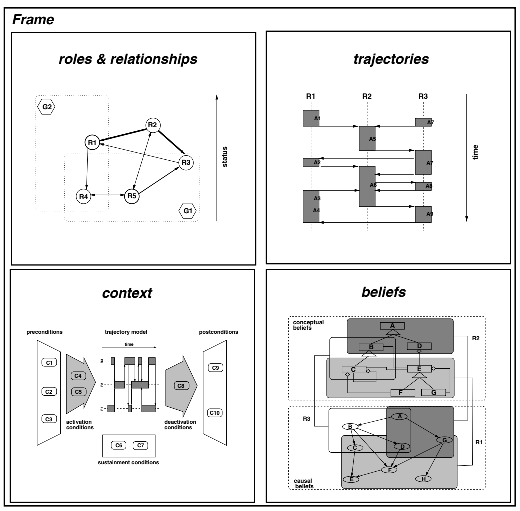

Figure 5: Integrated frame data structure.

oriented sense) and also by semantic relations (such as "F is an alternative to G", "F is a variant of G by sharing the same set of roles" etc.), and *histories*, that relate a frame to previous or subsequent frames by recording the modifications performed when deriving new frames ("F was derived from G by adding precondition C to its context model") are both captured by building up *frame repositories*. These are databases that comprise various frames in the form shown in Figure 5 linked to each other which are used by the agent (as information that is local to its mental processes) to manage the framing process described in Section 4.

3.2 Private attributes

If frames are to aid in the process of *exploiting* interaction experience, common attributes that represent shared interaction knowledge must be supplemented with local information about the individual experiences and evaluations of the agent using them. This is the purpose of private attributes, which basically contain *status* slots for each of the common attributes (role assignment status, trajectory status, activation status and belief status). All of these contain *mappings* for all facts in the respective common attribute and *assessments* concerning the private evaluations of the current state of affairs. To keep things simple, we will not extend the notation of Figure 5, since status data can be simply added to the four slots already introduced.

As for the concrete processes by which the resulting values of private attributes are determined, they are the result of the *framing* process in InFFrA that is described in more detail in the next section.

4. FRAMING AGENT ARCHITECTURE

Designing a social reasoning architecture based on the notion of *framing* means both describing how frames are constructed by an agent and how they are used in interactions. Framing is a very complex activity that involves (1) tracking the enactment of activated frames, (2) choosing whether to retain the current frame or to change frame when appropriate, (3) modifying frame knowledge with experience and (4) relating these three activities to one's private goals in order to make them part of individually rational decision-making.

The full architecture is shown in detail in Figure 6. To describe its features, we first introduce the data structures it employs and then the top-level processing steps that operate on these data structures[3].

The data structures that are used to perform these steps are the following:

- the *active frame* (the unique frame currently activated),
- the *perceived frame* (a frame-wise interpretation of the currently observed state of affairs),
- the *difference model* (containing the differences between perceived frame and active frame),
- the *trial frame* (the current *hypothesis* when alternatives to the current frame are sought for),
- and the *frame repository*, a (suitably organised) frame database used as a *hypothesis space*.

The top-level view of the framing process can be described by the following steps that an agent has to perform in each reasoning cycle:

1. Situation interpretation: The *situation interpretation module* obtains the *perceived frame* and incoming percepts as inputs. It outputs a frame that is used as a *descriptive* model of the current situation for matching purposes. Its task is to record ongoing interaction in terms of interaction frames, i.e. to distinguish between the trajectory itself, actor attributes that determine roles and relationships, context-relevant percepts and the beliefs of parties involved.

2. Matching: During *frame matching*, the current interaction situation as represented by the *perceived frame* (a descriptive model of interaction "as is") is compared with the *active frame*[4] (a normative picture of what the interaction "should be like"). The *difference model* is generated as

[3]Due to space limitations, we are not able to present details for each component of the architecture here. We refer the interested reader to [8].

[4]In the process of re-framing, the *perceived frame* is compared with *trial frames* that are "mock-activated".

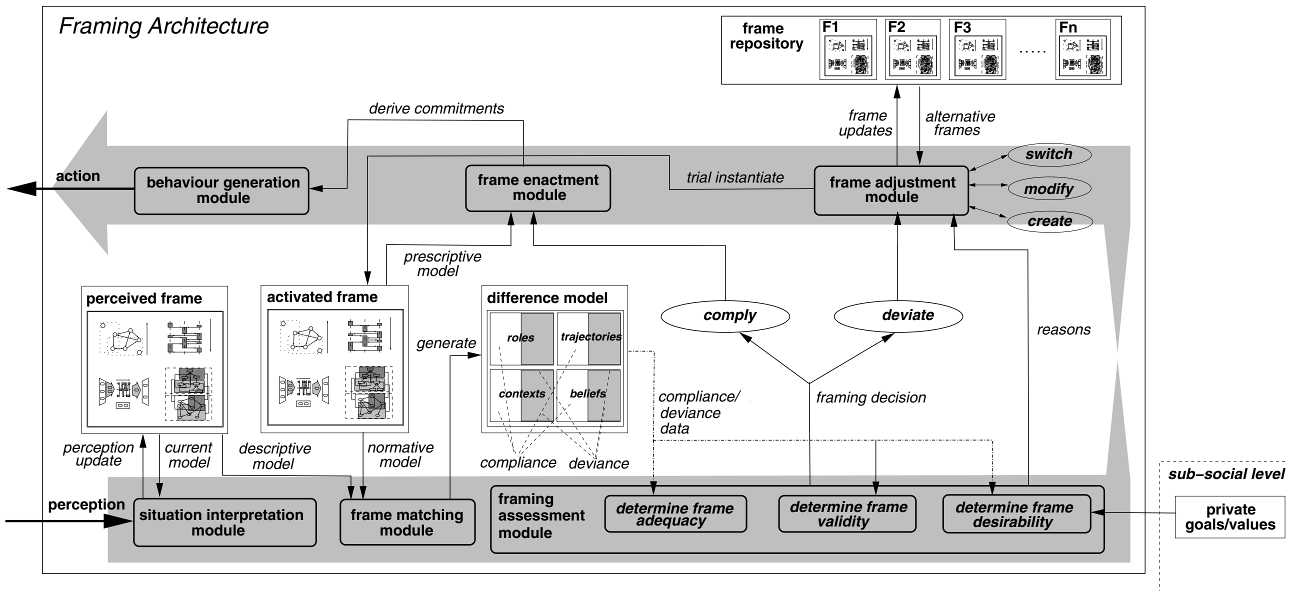

Figure 6: Detailed view of the framing-based agent architecture. The main line of reasoning between perception and action (shown as a shaded arrow) captures both the sub-processes involved and the temporal order in which they occur.

a result of this process, which contains lists of observations that *conform* with the active frame and of percepts that *deviate* from the expected course of action.

3. Assessment: The *frame assessment module* evaluates data obtained from the difference model with respect to three measures: *frame adequacy*, *frame validity* and *frame desirability*.

Adequacy reflects the degree to which the context model is satisfied by the current situation, which serves two purposes: (i) failure to meet context preconditions and sustainment conditions jeopardises correct execution of the actions prescribed by the frame and (ii) occurrence of activation/deactivation conditions implies adoption/abandonment of the frame.

Validity expresses the degree to which the interaction process matches the trajectory data. Mainly, it is used to infer whether the interacting parties meet the expectations induced by the frame, i.e. whether they are "doing the right thing". The result of assessing frame validity should be a detailed description of who fails to comply with which expectation for what reason. Obviously, if this description proves the active frame to be a false interpretation of what is going on, re-framing must be attempted.

Desirability, finally, reflects the use of the current frame with respect to the agent's private goals. Postconditions in the context of the frame as well as states of affairs that occur during the enactment of the trajectory must be desirable for the agent – otherwise, it should find a better alternative[5].

4. Framing decision: If the current frame seems appropriate according to the results of *frame assessment*, the process continues with step 6 (*comply*), else with step 5 (*deviate*). In the case of *deviate*, the reasons for the re-framing decision are output to the adjustment model to better guide the search for a new frame.

5. Adjustment/Re-Framing: *Frame adjustment* occurs when the agent has decided to cancel activation of the current frame. The adjustment module suggests *trial frames* using (1) the reasons for re-framing, (2) the frame repository data and (3) operators that allow for *switching to*, *modifying* and *creating new* frames. These trial frames are then used as "mock instances" of the *active frame*, and the framing process continues with step 2.

6. Enactment: The *frame enactment module* generates commitments for the agent that result from the employed frame. It outputs directives to the *behaviour generation module* which inform the agent about social obligations, permissions and prohibitions that it should respect.

7. Behaviour generation: Current social constraints are obtained from the *frame enactment module* and influence the action choices of the agent. Usually, action choices made by the agent at a sub-social level are simply overruled by these social-level decisions. However, alternatives to this strict overriding of local reasoning can be conceived of, and so we include this module to cater for richer models of combining individual with social choices.

The intuition behind this framing procedure is a very simple one: as long as the activated frame seems appropriate, it is maintained and influences the behaviour generating (sub-social) processes that the agent employs. If it fails to match the needs of the situation or those of the agent, alternatives are sought for, and if none exist, frames are adapted until promising alternatives are found. These are iteratively instantiated as "mock frames" and tested against current conditions. If a suitable alternative has been found, it is

[5]This is not to say that only desirable frames will be activated by the agent. Of course, circumstances may require the opposite, e.g. due to long-term commitments, reciprocal relationships, access to resources, etc. Suggesting a view of self-interested rational agents should not insinuate that the agent's rationality is not *socially bounded* in practice.

activated, otherwise, we try again.

This logic is very similar to that used in case-based reasoning methods [6], InFFrAmight in fact be called a *case-based interaction learning method* for knowledge-based deliberative agents.

With respect to our outlook on autonomy, one last thing to note is that, once activated, a frame *does* limit individual agent rationality and deliberation, but only if the agent wishes to submit herself to it.

In the following evaluation, we exemplify how this architecture provides useful guidance for identifying relevant issues in the process of analysing and developing framing multiagent systems. This evaluation will also explain how concrete implementations can further elaborate certain aspects such as the prevention of infinite trial instantiation (loops between steps 2 and 5), the combination of adequacy, validity and desirability measures, and goal-oriented generation of new frames in the process of re-framing.

5. EVALUATION

5.1 Application: Strategic Opponent Classification in Multiagent Games

In order to evaluate the adequacy of our framework, we apply it to the prototype of an opponent classification system for iterated multiagent games that has been recently developed in our research group [9]. In this system, agents moving on a toroidal grid play a fixed number of Iterated Prisoner's Dilemma [4, 7] games whenever they happen to be in the same caret with some other agent (when more than two agents meet, every player plays against every other player). The goal of the system is to extend the model-based learning method *US-L** proposed by Carmel and Markovitch [3, 2] (that is based on learning opponent behaviour in terms of a deterministic finite automata (DFA)) by classification capabilities, so that in large-scale MAS agents need not build a model of *every* opponent but only of the different *classes* of opponent behaviour that exist.

For lack of space, it is not possible to lay out the details of the opponent classification MAS here (they can be found in [9]). To give an idea of the most important aspects of the method, however, we should mention that each agent maintains a (variably-sized, bounded) set of opponent classes

$$\mathcal{C} = \{c_i = \langle A_i, Q_i, S_i \rangle | i = 1, \dots k\}$$

where each class consists of (i) a DFA A_i that models the behaviour of opponents in c_i, (ii) a Q-table [10] Q_i to learn optimal strategies against A_i (the state space of the Q-table is the state space of A_i, entries are updated using game rewards); (iii) a set of samples (recent fixed-length sequences of game moves of both players) S_i with which A_i is trained (these are collected whenever the modelling agent plays against class c_i). Further, a similarity measure $\sigma : \mathit{Agents} \times \mathcal{C} \rightarrow [0; 1]$ between adversaries and classes is maintained, as well as a membership function $m : \mathit{Agents} \rightarrow \mathcal{C}$ that describes which opponent pertains to which class.

After a sequence e of games has been played with opponent a during an "encounter", the modelling agent updates the sample set S_i for $c_i = m(a)$ ($m(a)$ is initially undefined ($\perp$)) and adapts A_i if it fails to predict e correctly. Also, the values in Q_i are updated with payoffs received during e, so that an optimal strategy is learned over time. Then,

Algorithm 1 Top-level algorithm

1: ***inputs:*** Adversary a, Encounter e, Integer k
2: ***outputs:*** Set $\mathcal{C}$, Membership function m
3: **begin**
4: $\forall c \in \mathcal{C}.\sigma(a, c) \leftarrow \frac{correct(a,c)}{all(a)}$
5: **if** $m(a) = \perp$ **then**
6:   $m(a) \leftarrow \mathit{BestClass}(a, e, \mathcal{C}, k, 1)$
7: **else**
8:   **if** $m(a)$ doesn't predict e correctly **then**
9:     **if** $S(a, m(a)) \leq \delta \vee m(a)$ is very stable **then**
10:       $m(a) \leftarrow \mathit{BestClass}(a, e, \mathcal{C}, k, \rho_1)$
11:     **end if**
12:     $c' \leftarrow \mathit{BestClass}(a, e, \mathcal{C}, k, \rho_2)$
13:     **if** $c' \in \mathcal{C} \wedge c'$ is very stable **then**
14:       $m(a) \leftarrow c'$
15:     **end if**
16:     Om-Learn$(m(a), e)$
17:     **if** $m(a)$ has been modified **then**
18:       $\forall m(a') \neq m(a).\sigma(a', m(a)) \leftarrow 0$
19:     **end if**
20:     $\mathcal{C} \leftarrow \mathcal{C} - \{c | \forall a'.m(a') \neq c\}$
21:   **end if**
22: **end if**

the top-level classification procedure is called as outlined in Algorithm 1.

The algorithm strongly relies on the definition of a function $\mathit{BestClass}(a, e, \mathcal{C}, k, \rho)$ which retrieves the most appropriate class for opponent a given the current encounter e. The function attempts to find a class in $\mathcal{C}$ that matches e with at least similarity ρ. If no such class can be found, a new class is created, unless $|\mathcal{C}| = k$, i.e. the upper bound on the number of admissible opponent classes has been reached. If $|\mathcal{C}| = k$, the constraint on ρ is dropped and the most similar class is returned.

The top-level algorithm begins with an update of all σ-values with respect to a in line 4 (similarities are always computed as the ratio between encounters with a correctly predicted by c $correct(a, c)$ and the total number of encounters with a $all(a)$); then, if a is an unknown agent (5–7), it is classified to any c that correctly predicts e (this is achieved by applying $\rho = 1$). In the counter-case (lines 8–20), nothing is done unless $m(a)$ has been modified because of e, i.e. if it has correctly predicted e. If $\sigma(a, m(a))$ falls below similarity threshold δ, or if $m(a)$ has been stable for a long time, a is re-classified to the maximally similar class, which has to be at least as similar as ρ_1 (lines 9–11). If no such class exists, the same strategy is followed as in line 6 (a new class is created if possible, else the most suitable class is selected).

If, on the other hand, similarity is larger than δ and the automaton is not highly stable, we still re-classify a, but only to highly similar ($\rho_2 \gg \rho_1$), highly stable classes, so that similar classes are merged in the long run (lines 12–15). The opponent model learning function Om-Learn is then called in line 16 to generate a model that is consistent with the new experience e.

Finally, if e caused modifications to $m(a)$, the σ-values for all agents not in $m(a)$ are reset to 0 (line 18) since nothing can be said about their similarity with $m(a)$, if the model for $m(a)$ has just changed; also, empty classes are erased from $\mathcal{C}$ (line 20).

As concerns action choices *during* encounters, the Q-table belonging to $m(a)$ is used (with additional Boltzmann exploration) if a has been classified before; if a is encountered for the first time, the most similar class is determined using σ-values after each move, and the move the Q-table of that class suggests is played.

Despite its simplicity, this classification heuristic is quite effective: when playing against n fixed strategies, it converges to n classes that can play optimally against arbitrary numbers of adversaries as long as these play one of the n strategies. If all agents use the above strategy, on the other hand, no discernible global behaviour emerges, agents make random action choices. Only if additional assumptions are made (e.g. if agents play TIT FOR TAT for a while themselves whenever the other exhibits random behaviour) can cooperation be established (these results are presented in more detail in [9]).

5.2 InFFrA-based Analysis

The above description has made clear how the algorithm works at a technical level. What is *not* clear, however, is its socio-cognitive dimension, an understanding of which is crucial for multiagent system analysis and design. What is therefore needed is a uniform framework in which this algorithm can be embedded and which clarifies the cognitive and social processes occurring within and between individual agents, as provided by InFFrA. In the following, we describe how the opponent classification MAS can be analysed and characterised in terms of InFFrA terminology.

The MAS just described clearly belongs to the class of systems suggested in Section 2, since agents in it record, organise and exploit regularities in interaction processes in a socially intelligent way.

Interaction Frames. The opponent classes created by the algorithm are interaction frames with a *trajectory model* (the automaton, a causal model of reaction to one's own actions) defined in terms of two roles whose relationship is one of mutual interdependence: one role is always fixed – the modelling agent itself; the other role is described by keeping track of all opponents who match it (this is done by the m-function). The *context model* is largely trivial: activation and de-activation conditions are simply "being in the same caret with an opponent" and "having finished l IPD games", pre- and sustainment conditions are empty. However, post-conditions in terms of reward expectations are represented by the Q-table. As for *beliefs*, these are implicit to the architecture: both agents know their action choices (capabilities), both know the game has a fixed length, both know that the other's choices matter. *Links* exist implicitly between all frames since they are all exclusive alternatives to each other: they share role sets, belief models and context (apart from post-conditions) and are tailored for the same kind of interaction (since there occurs only one type of interactions); in fact, the frames can only be distinguished by DFA-, Q-table-, S_i- and σ-data.

As for *private attributes*, these are of course the actual entries in the Q-table (own payoffs cannot be perceived by opponents) and the samples in S_i-sets that reflect the past experiences with that frame (*histories*). Also, a role assignment takes place whenever encountering an agent by using A_i to predict its behaviour, and state changes in the DFA track trajectory status during play. Tracking context is trivial except for the Q-update, but the update of the σ-function (line 4 in Algorithm 1) as well as re-setting their values in case of DFA modification (line 18) track framing experience across frames, and they actually manipulate *all* frames' private attributes simultaneously. More particularly, the *difference model* that is represented by σ-values is constantly computed for *all* frames with respect to the current interaction situation (cf. below).

These frames are certainly based on common knowledge (as far as common attributes are concerned) and they are grounded in experience; also, the DFA models generalise from individual actors and actions by re-constructing mental states of opponents, and together with Q-learning they are instrumentalised by the modelling agent to optimise payoff performance. Thus, their use covers the criteria listed in Section 3.

Framing. To illustrate how the opponent classification MAS can be viewed in terms of a framing agent architecture, we re-trace the top-level processing steps described in Section 4 and identify the respective algorithmic components in the opponent classification MAS. At the same time, we use InFFrA to identify *weaknesses* and *advantages* of the MAS to underline its usefulness for improving existing systems.

We need to distinguish between (i) the case in which the current opponent a has been encountered before and (ii) the case in which we are confronted with an unknown adversary.

Case (i): The *matching process* occurs at the start of every encounter; $m(a)$ is chosen (blindly) as the most appropriate frame from the repository and is activated.

This choice is then never altered during the encounter and this is a first disadvantage of the system, because no frame assessment and adjustment occurs *during* encounters, thus limiting the adaptability of the modelling agents severely within the current interaction.

The *situation interpretation module* records the current sequence of moves (the *perceived frame*), stores it in $S_{m(a)}$ and updates the entries in the Q-table according to recent payoffs. The *frame matching module* updates similarity values (line 4 in Algorithm 1) for *all* frames with respect to a. Compared to the InFFrA-intuition, this is a much more complex matching activity, since it compares the difference model with all classes, so that the lack of framing assessment is partly made up for by adding complexity at the frame matching stage.

As mentioned, *frame assessment* and *re-framing* occur only *after* the encounter: frame validity is assessed according to whether the current sequence of opponent moves is understood by the DFA in $m(a)$ or not. Here, we observe a second drawback of the prototype: adequacy and desirability assessment is clearly under-developed, since neither *consistency* of Q-values nor the *usefulness* (e.g. expected future payoff with that class) of frames is taken into account. For example, the classification heuristic would not be able to cope with types of opponents whose actions have *different* utility outcomes for the modelling agent. If, e.g., own payoff matrix entries differed across opponents, those opponents would still end up being classified identically if they perform identical actions. Also, since the agent has no choices regarding partner selection, it does not make sense to weigh the desirability of entire Q-tables against each other.

As a consequence, the *framing decision* also has only effects on *future* encounters with the same agent. It depends on the simple criterion of whether the DFA of c has just been modified or not. If so, the *frame adjustment module* comes into play: it potentially re-classifies a, creates a new class for it and resets similarity values for non-members of c; at the same time, it seeks to retain highly stable classes and to merge similar classes in the long run.

This is undoubtedly the most elaborate component in the opponent classification MAS, and it nicely illustrates the possibilities of a long-term organisation of interaction experience, especially because, in a boundedly rational manner, it tries to distinguish between frames only where necessary. *Trial instantiation*, on the other hand, is very simple: it can be trivially reduced to using the maximally (and highly) similar candidate frame as the new value for $m(a)$, because similarities between a and all classes are constantly tracked.

Frame enactment is performed by tracking opponents' actions in the current DFA and by selecting the next move according to the Q-table. Then, since there are no other reasoning levels to compete with, *behaviour generation* is straightforward.

In this enactment stage, a third shortcoming can be identified that probably explains the lack of structure in interaction among modelling agents described at the end of Section 5.1, namely the fact that the frames impose no restrictions on the action selection mode of the modelling agent itself – in fact, the trajectory represented by the DFA prescribes only the actions of the opponent, and the modelling agent is merely optimising its behaviour towards the opponent. Therefore, since no agent feels it should comply with some more specific pattern of behaviour, no recurring efficient interaction patterns can occur.

Case (ii): In the case of unknown opponents, frame assessment and re-framing is implemented as in the previous case. The differences lie in matching and in making framing decisions, which occurs after each *round* of the encounter (and not only after the entire encounter), since after each round, the modelling agent activates the most similar class with respect to the current sequence of moves and uses this class for enactment decisions (according to the respective Q-table).

Again, this illustrates the implementation of the bounded rationality principle: the framing effort is in this case much greater (and adheres much more to InFFrA requirements), given that interactions with unknown agents are much riskier than those with known adversaries. This suggests that an extension of the opponent classification heuristic that also allows for re-framing *during* encounters in case (i) might increase agent performance, yet at a greater computational cost.

This evaluation provides evidence for the practical use of InFFrA as a conceptual framework that supports the analysis of existing MAS by decomposing social reasoning algorithms into functional components that have to fulfil certain requirements. Starting from these requirements, we can then identify advantages and shortcomings of the analysed systems and suggest improvements. In that, InFFrA itself is designed to match the needs of a particular class of MAS which we find under-represented in current MAS research, namely systems consisting of agents whose social intelligence is based on the ability to record, organise and exploit regularities observed in social interaction processes.

6. CONCLUSIONS

This paper has presented the InFFrA method for viewing multiagent systems as societies of agents that engage in *framing*, i.e. in recording, organising and strategically employing regularities in micro-social interactions. We have based this method on a conceptual social reasoning architecture that is built upon the notion of *interaction frames*, i.e. units of interaction knowledge that can be used to structure the social world for the reasoning agent, as proposed by the sociological theory of Erving Goffman [5].

The benefits of our approach have been illustrated using a prototypical strategic opponent classification MAS for learning in games. Apart from showing how an agent architecture can be modelled and analysed in terms of the InFFrA terminology and intuitions, we have given examples for potential improvements of this existing MAS based on the principled, socio-centric view InFFrA proposes.

Future challenges include (i) exploring the usage of frames in richer scenarios, particularly with a focus on the pragmatics of communication, (ii) developing formal representations for the concepts introduced here, (iii) investigating the co-evolution of subjective frame models, the emergence of stable, intra-subjectively valid frames and (iv) developing concrete architectures in order to achieve efficient, flexible coordination among self-interested but socially conscious agents.

We believe that in current MAS research, the analysis of the mutual dependence between cognitive agent processing of interaction experience and the evolution of social regularities (which lead to phenomena such as (self-)organisation, the evolution of norms and institutions, group formation etc.) is still at its beginning. Using InFFrA as a tool for analysing and modelling precisely this relationship certainly leads to a better understanding of the problems associated with it.

7. ACKNOWLEDGEMENTS

This work has been supported by the DFG under contracts no. Br 609/11-1 and Br 609/11-2 under the "Socionics" priority research program. We are indebted to Michael Schillo and Matthias Nickles for useful comments on earlier versions and to Kai Paetow and Marco Schmitt for providing a sociological perspective.

8. REFERENCES

[1] W. Brauer, M. Nickles, M. Rovatsos, G. Weiß, and K. F. Lorentzen. Expectation-Oriented Analysis and Design. In *Proc. AOSE'2001*, volume 2222 of *LNCS*. Springer-Verlag, Berlin, 2001.

[2] D. Carmel and S. Markovitch. Learning and using opponent models in adversary search. Technical Report 9609, Technion, 1996.

[3] D. Carmel and S. Markovitch. Learning models of intelligent agents. In *Proc. AAAI'1996*, pages 62–67, Menlo Park, CA, 1996. AAAI Press/MIT Press.

[4] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, Cambridge, MA, 1991.

[5] E. Goffman. *Frame Analysis: An Essay on the Organisation of Experience*. Harper and Row, New York, NY, 1974.

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

[7] R. D. Luce and H. Raiffa. *Games and Decisions*. John Wiley & Sons, New York, NY, 1957.

[8] M. Rovatsos. Interaction frames for artificial agents. Technical Report Research Report FKI-244-01, Technical University of Munich, 2001.

[9] M. Rovatsos and M. Wolf. Towards Social Complexity Reduction in Multiagent Learning: the AdHoc Approach. Technical Report SS-02-02, AAAI Press, Stanford, CA, 2002.

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