Communication Systems: A Unified Model of Socially Intelligent Systems*

Matthias Nickles, Michael Rovatsos, Wilfried Brauer, and Gerhard Weiß

Department of Informatics, Technical University of Munich
85748 Garching bei München, Germany,
{nickles,rovatsos,brauer,weissg}@cs.tum.edu

Abstract. This paper introduces *communication systems* (CS) as a unified model for socially intelligent systems. This model derived from sociological systems theory, combines the empirical analysis of communication in a social system with logical processing of social information to provide a general framework for computational components that exploit communication processes in multiagent systems. We present an elaborate formal model of CS that is based on expectation networks and their processing. To illustrate how the CS layer can be integrated with agent-level expectation-based methods, we discuss the conversion between CS and interaction frames in the InFFrA architecture. A number of CS-based applications that we envision suggest that this model has the potential to add a new perspective to Socionics and to multiagent systems research in general.

1 Introduction

Traditional attempts to model the semantics of agent communication languages (ACLs) are mostly based on describing mental states of communicating agents [2, 3, 7, 19] or on observable (usually commitment-based) social states [6, 14, 20]. However, both these views fail to recognise that communication semantics evolve during operation of a multiagent system (MAS), and that the semantics always depend on the view of an observer who is tracking the communicative processes in the system. Yet this is a crucial aspect of inter-agent communication, especially in the context of open systems in which a predetermined semantics cannot be assumed, let alone the compliance of agents' behaviour with it.

In [8] we have therefore – influenced by sociological systems-theory [9] – introduced *expectations* regarding observable *communications* as a universal means for the modelling of emergent sociality in multiagent systems, and in [15], we have presented – influenced by actor-oriented, socio-cognitive theories [5, 11] – a formal framework for the semantics of communicative action that is *empirical*, *constructivist* and *consequentialist* in nature and analysed the implications of this model on social reasoning from an agent perspective. We suggested that recording observations of message exchange among agents in a multiagent system (MAS) *empirically* is the only feasible way to capture the meaning of communication, if no *a priori* assumptions about this meaning

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

can be made. Being empirical about meaning naturally implies that the resulting model very much depends on the observer's perspective, and that the semantics would always be the semantics "assigned" to utterances by that observer, hence this view is inherently *constructivist*. Since, ultimately, no more can be said about the meaning of a message in an open system than that it lies in the set of expected consequences that a message has, we also adopt a *consequentialist* outlook on meaning.

In this paper, we integrate and extend upon these views that were strongly influenced by different sociological views. We present a detailed framework for the formal description of socially intelligent systems based on the universal, systems-theoretical concept of *communication systems*, which subsumes structure-oriented expectation modelling on the one hand, and the modelling of cognitive, goal-oriented social knowledge of active agents on the other.

In the terminology of sociological systems theory, communication systems are systems that consist of interrelated communications which "observe" their environment [9]. For the purposes of this paper, we will use the term "communication system" (CS) to denote computational entities that process empirical information about observed communication[1] and use this information to influence the behaviour of the underlying system. Their distinct features are (i) that they only use data about communication for building models of social processes, the underlying assumption being that all relevant aspects of social interaction are eventually revealed through communication, and (ii) that, different from a passive observer, the may take action in the system to influence its behaviour; in other words, there is a feedback loop between observation and action, so that a CS becomes an autonomous component in the overall MAS.

Note, however, that CSs need not necessarily be (embedded in) agents. Although their autonomy presumes some agentification, their objectives need not be tied to achieving certain goals in the physical (or pseudo-physical simulation) world as is the case with "ordinary" agents. Thus, they are best characterised as components used to (trans)form expectations (regardless of how these expectations are employed by agents in their reasoning) and which are autonomous with respect to how they perform this generation and modification of expectations.

Our hypothesis regarding the Socionics endeavour [10] is that its main contribution lies in the construction of appropriate communication systems for complex MAS, or, to take it to the extreme, we might summarise this insight as

$$\textit{Socionics} = \textit{empirical communication analysis} \; + \; \textit{rational action}$$

because the CS viewpoint extends the traditional outlook on MAS taken in the field of distributed artificial intelligence (DAI). Thereby, the "semantics" aspect mentioned above plays a crucial role, because meaning lies in the sum of communicative expectations in a system, and CS capture precisely these expectations and how they evolve.

The remaining sections are structured as follows: We start by introducing expectation networks in section 2, which constitute our formal model for describing communicative expectations. Then, we formally define communication systems and their

[1] I.e., our CS realises some kind of *second-order observer* in terms of sociological systems theory.

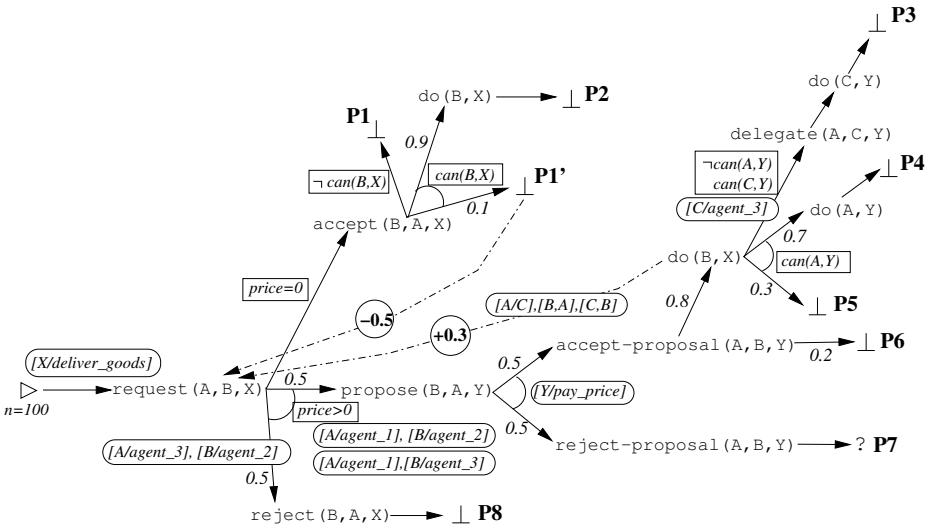

Fig. 1. An expectation network. Nodes are labelled with message templates in `typewriter` font and the special symbols "▷", "⊥" and "?"; they are connected by (solid) cognitive edges labelled with numerical expectabilities in *italic* font, or (dashed) normative edges with round circles containing a numerical "force" value in **bold** face. Substitution lists/conditions belonging to edges appear in rounded/edged boxes near the edge. If neighbouring edges share a condition this is indicated by a drawn angle between these edges. The path labels **P1** to **P8** do not make part of the notation and are simply used to refer to paths in the text.

semantics (section 3). As an example of how CS can be used to model socially intelligent systems, we discuss the conversion between the social reasoning architecture InFFrA and CS in section 4. Section 5 discusses possible applications of the CS, and section 6 concludes.

2 Expectation Networks

Expectation networks (ENs) [8] are the data structures on which communication systems operate. They capture regularities in the flow of communication between agents in a system by connecting message templates (nodes) that stand for utterances with each other via links (edges) which are labelled with (i) weights (ii) a logical condition and (iii) lists of variable substitutions. Roughly speaking, the semantics of such a labelled edge is that "if the variables in the messages have any of the values in the substitution lists, and the logical condition is currently satisfied, then the weight of the edge reflects the frequency with which a message matching the label of the target node follows utterance of a message matching the label of the source node of the edge." Before presenting a full definition of ENs, we have to introduce some basic notions and notation we use, and to make certain auxiliary definitions and assumptions. The example network in figure 1 will be used throughout the discussion of ENs to illustrate the purpose of definitions and assumptions.

2.1 Basics

A central assumption that is made in ENs is that observed messages may be categorised as *continuations* of other communications, or may be considered the start of a new interaction that is not related to previous experience. So an edge leading from m to m' is thought to reflect the *transition probability*, i.e. the frequency of communication being "continued" from the observer's point of view. Usually, continuation depends on temporal and spatial proximity between messages, but it might also be identified through a connection about "subject", or, for example, though the use of the same communication media (m' was shown on TV after m was shown some time earlier on).

Apart from "ordinary" node labels, we use three distinct symbols "$\triangleright$", "$\bot$", and "?". "$\triangleright$" is the label occurring only at the root node of the EN. Whenever a message is considered a "non-continuation" of previous sequences (i.e. the start of a new interaction), it is appended to this "$\triangleright$"-node. Nodes labelled with "$\bot$" denote that a communication ended after the label of the predecessor of this node was observed. The label "?", finally, indicates that there exists no expectation regarding future messages at this node. Nodes with such "don't know" semantics are usually messages that occur for the first time – the observer noes nothing about what will happen after them being uttered. To define the syntactic details of EN, we have to introduce formal languages $\mathcal{L}$ and $\mathcal{M}$ used for logical expressions and for message templates, respectively. $\mathcal{L}$ is a simple propositional language consisting of atomic propositions $Statement = \{p, q(X, s), \ldots\}$ potentially containing (implicitly universally quantified) variables (for which we use capitalised letters, e.g. X), of the usual connectives $\vee$, $\wedge$, $\Rightarrow$ and $\neg$, the logical constants "true" and "false", and braces $()$ for grouping sub-expressions together (the language is formally given by the grammar in table 1). We write $\models \varphi$ if φ. A *knowledge base* $KB \in 2^{\mathcal{L}}$ can be any finite set of formulae from $\mathcal{L}$. For simplicity, we will often write $KB \models \varphi$ to express $\models (\wedge_{\varphi' \in KB} \varphi') \Rightarrow \varphi$.

As for $\mathcal{M}$, this is a formal language that defines the message patterns used for labelling nodes in expectation networks. Its syntax is given by the grammar in table 1. Messages observed in the system (we write $\mathcal{M}_c$ for the language of these *concrete* messages) can be either physical messages of the format $\mathtt{do}(a, ac)$ where a is the executing agent and ac is a symbol used for a physical action, or a non-physical message $performative(a, b, c)$ sent from a to b with content c. (Note that the terminal symbols used in the $Agent$ and $PhysicalAction$ rules are domain-dependent, and that we take the existence of such symbols for granted.) However, the message labels of type $MsgPattern$ used in expectation networks may also contain variables for agents and physical actions (though not for performatives). As we will soon show, this is useful to generalise over different observed messages by adding a variable substitution to each node. The content c of a non-physical action, finally, is given by type $LogicalExpr$. It can either be (i) an atomic proposition, a (ii) message pattern or physical action term, (iii) an expectation network, or (iv) a logical formula formed out of these elements. Syntactically, expectation networks ($Graph$) are represented as lists of edges (m, p, n, c, l) where m and n are message patterns, p is a transition probability from m to n, c is a logical condition, l is a list of variable substitutions. The meaning of these will be clarified once the full definition of expectation networks has been presented.

$$
\begin{aligned}
Var &\rightarrow X \mid Y \mid Z \mid \ldots \\
AgentVar &\rightarrow A_1 \mid A_2 \mid \ldots \\
PhysicalActVar &\rightarrow X_1 \mid X_2 \mid \ldots \\
Expect &\in [0;1] \\
Agent &\rightarrow agent\_1 \mid \ldots \mid agent\_n \\
Head &\rightarrow it\_rains \mid loves \mid \ldots \\
Performative &\rightarrow accept \mid propose \mid reject \mid inform \mid \ldots \\
PhysicalAction &\rightarrow move\_object \mid pay\_price \mid deliver\_goods \mid \ldots \\
Message &\rightarrow Performative(Agent, Agent, LogicalExpr) \\
&\quad \mid do(Agent, Agent, PhysicalAction) \\
MsgPattern &\rightarrow Performative(AgentTerm, AgentTerm, LogicalExpr) \\
&\quad \mid do(AgentTerm, AgentTerm, PhysicalActTerm) \\
&\quad \mid \triangleright \mid \bot \mid ? \\
PhysicalActTerm &\rightarrow PhysicalActVar \mid PhysicalAction \\
AgentTerm &\rightarrow AgentVar \mid Agent \\
LogicalExpr &\rightarrow (LogicalExpr \Rightarrow LogicalExpr) \mid (LogicalExpr \vee LogicalExpr) \\
&\quad \mid (LogicalExpr \wedge LogicalExpr) \mid \neg LogicalExpr \\
&\quad \mid Statement \\
Statement &\rightarrow Head \mid Head(TermList) \mid \text{true} \mid \text{false} \\
TermList &\rightarrow TermList, Term \mid Term \\
Term &\rightarrow Var \mid AgentTerm \mid MsgPattern \mid Graph \\
EdgeList &\rightarrow (MsgPattern, Expect, MsgPattern, LogicalExpr, SubstList)\ EdgeList \mid \varepsilon \\
Graph &\rightarrow \langle EdgeList \rangle \\
SubstList' &\rightarrow SubstList'\ Subst \mid \varepsilon \\
SubstList &\rightarrow \langle SubstList' \rangle \\
Subst &\rightarrow [AgentVar/Agent] \mid [PhysicalActVar/PhysicalAction] \\
&\quad \mid [Var/Term]
\end{aligned}
$$

Table 1. A grammar for messages, generating the languages $\mathcal{M}$ (the language of message patterns, using $MsgPattern$ as starting symbol), $\mathcal{M}_c$ (the language of concrete messages, using $Message$ as starting symbol) and the logical language $\mathcal{L}$ (using $LogicalExpr$ as starting symbol.

2.2 Normative and Cognitive Edges

A distinct feature of ENs is that their edges fall into two categories, *cognitive* and *normative* edges. A cognitive edge e (also called observation edge) denotes a correlation in observed communication sequences. Usually, its expectability $Exp(e) \in [0;1]$ reflects the probability of $target(e)$ occurring shortly after $source(e)$ in the same communicative context (i.e. in spatial proximity, between the same agents, etc.). Although expectability values need not directly reflect the frequency of a continuation that matches

the source and target node labels (in the context of the path leading to the source node), they should somehow correlate with it.

Normative edges, on the other hand, represent knowledge about correlations between messages that does not have its origins in observation[2]. In particular, normative edges may link messages together that do not occur successively, and therefore the EN is not a tree if we include normative edges in the edge set. Therefore, they are not annotated with degrees of expectation, but with a numerical assessment of the "normative force" they exert. This value $Fc(e,n) \in \mathbb{R}$ is thought to increase or decrease the probability of $target(e)$ whenever $source(e)$ occurs *regardless* of any observed correlation between $source(e)$ and $target(e)$. The second argument n used to compute this force is the time that passed since last observing a message that matched $source(e)$. Obviously, its absolute value should decrease with time passing, and it should become zero after some time, i.e.

$$\forall n_1, n_2 \in \mathbb{N}. n_2 > n_1 \Rightarrow |Fc(e,n_2)| < |Fc(e,n_1)| \quad (1)$$

$$\exists n_0 \in \mathbb{N}. \forall n > n_0. Fc(e,n) = 0 \quad (2)$$

Computation of changes in the impact of a normative edge necessitates, of course, keeping track of the time $\tau(v)$ that has passed since a message was observed that matched the label of node v. Note that, usually, the function definition for Fc will be identical for all normative edges in a EN except for initial values $Fc(e,0)$. Computation of changes in the impact of a normative edge necessitates, of course, keeping track of the time $\tau(v)$ that has passed since a message was observed that matched the label of node v. To illustrate the usefulness of normative links, consider the following paths in the EN of figure 1:

$$\textbf{P2} : \texttt{request}(A,B,X) \rightarrow \texttt{accept}(B,A,X) \rightarrow \texttt{do}(B,X)$$

$$\textbf{P1'} : \texttt{request}(A,B,X) \rightarrow \texttt{accept}(B,A,X) \rightarrow \bot$$

These represent two possible runs in a conversation in which A wants B to perform some physical action X. In the first case, B executes the promised action X, in the second he does not. What normative edges would be reasonable to add? Possible suggestions would be

$$e_1 = (\texttt{do}(B,X), \texttt{request}(A,B,X)), Fc(e_1,n) > 0$$

$$e_2 = (\bot, \texttt{request}(A,B,X)), Fc(e_2,n) < 0$$

$$e_3 = (\texttt{accept}(B,A,X), \texttt{do}(B,X)), Fc(e_3,n) > 0$$

$$e_4 = (\texttt{do}(B,X), \texttt{request}(A,B,X)), Fc(e_4,n) > 0,$$
$$subst(e_4) = \langle [A/C], [B/A], [C/B] \rangle$$

e_1/e_2 would increase/decrease the probability of future requests from the same agent A to B depending on whether the conversation "worked out" for him or not.

[2] In practice, normative edges can be the product of "very long term" observation of communication. However, we will uphold this clear distinction between cognitive and normative here because normative knowledge can probably not be derived in the scope of observation we assume here. Introducing a gradual sense of normativity and cognitivity for edges would be another option that might be investigated in the future.

Note that e_2 (which is actually present in the EN of figure 1) will only have negative consequences for the "reputation" of B if he promised to perform X – if he used `reject` instead (path **P8**), he would not be sanctioned, and this would effectively strengthen the normative character of `accept`. Edge e_3 induces an increase on the probability of `do` once an agent has accepted, i.e. it suggests some semantic relationship between accepting to do something and then actually doing it. If an observer is using e_3, this means he is implementing a norm which does not depend on how often actually agents were observed to perform some task that they had previously accepted to take over. Finally, e_4 represents a relationship between messages more complex still: it suggests that if agent B does what A requested of him, this increases the probability of B asking something from A in reverse ($subst(e)$ being the list of substitutions stored with edge e). This example nicely illustrates the different uses of normative edges in particular in their different functions as "expectability manipulators" in prospective and retrospective ways.

2.3 Edge Conditions

As a final ingredient to network edges, we briefly discuss edge conditions. The idea is that this condition should further define the scope of validity to cases in which a formula can be shown to hold using the observer's knowledge base. So, if $\varphi = cond(e)$, then e is only relevant iff $KB \models \varphi$. Although this idea is straightforward, we have to make some additional assumptions regarding these conditions to facilitate the definition of EN semantics.

First of all, the sum of expectabilities of all cognitive out-edges of a node should be one for any state (i.e. set of believed facts) of the knowledge base. In other words, the condition

$$\forall v \sum_{e \in out(v), KB \models cond(e)} Expect(e) = 1 \tag{3}$$

should hold. This can be ensured, for example, by guaranteeing that the following condition holds through appropriate construction rules for the EN. Assume the outgoing links $out(V)$ of every node v are partitioned into sets $O_1, O_2, \ldots O_k$ where the links' expectabilities in each O_i are non-negative and sum to one[3]. Now let all edges in O_i share the same edge condition, i.e. $\forall i \exists \varphi \forall o \in O_i.(cond(o) = \varphi)$ and define $cond(O_i)$ as precisely this shared condition φ. (The O_i sets are precisely those sub-sets of $out(v)$ connected by a drawn angle in figure 1.)

If we make sure that the outgoing links of every node are partitioned in this way, we can assign mutually exclusive conditions to them, i.e. ensure that

$$\forall i \neq j.cond(O_i) \wedge cond(O_j) \equiv \text{false} \quad \text{and} \quad \vee_i \, cond(O_i) \equiv \text{true} \tag{4}$$

This way, it is not only guaranteed that we can derive unambiguous probabilities directly from the *Expect* values, but also that we can do so for *any* knowledge base contents.[4]

[3] Formally, $out(v) = \cup_{1 \leq i \leq k} O_i$ and $\forall 1 \leq i < j \leq k.O_i \cap O_j = \emptyset$, and $\forall i \leq k.(\forall o \in O_i.Expect(o) \geq 0 \wedge \sum_{o \in O_i} Expect(o) = 1)$.

[4] This comes at the price of having to insert redundant edges in some situations. For example, insertion of a new edge e with $cond(e) = \varphi$ if $out(v) = \emptyset$ necessitates insertion of another edge e' with $cond(e) = \neg\varphi$.

2.4 Formal Definition

Having discussed all the prerequisites, we can now define ENs formally:

Definition 1. *An* expectation network *is a structure*

$$EN = (V, C, N, \mathcal{M}, \mathcal{L}, H, n, mesg, \tau, cond, subst, Expect, Fc)$$

where

- $|V| > 1$ *is the set of nodes,*
- $C \subseteq V \times V$ *are the* cognitive *edges of EN,* $N \subseteq V \times V$ *are its* normative *edges,*
- $G = (V, C)$ *is a tree called* expectation tree*,* $G = (V, C \uplus N)$ *is a graph,* $N \cap C = \emptyset$*,*
- $\mathcal{M}$ *is a* message pattern language*,* $\mathcal{L}$ *is a logical language,* $cond : C \uplus N \to \mathcal{L}$*,*
- $mesg : V \to \mathcal{M}$ *is the* message label *function for nodes with*
  - $mesg(v) = \triangleright$ *exactly for the root node of* (V, C)*,*
  - $\forall v \in V. \forall e, f \in out(v). \neg unify(mesg(target(e)), mesg(target(f)))$ *(target node labels of outgoing links never match),*
- $H \in \mathbb{N}$ *is a finite* communication horizon*,*
- $n \in \mathbb{N}$ *is the total number of* non-continuations*,*
- $Expect : C \to [0; 1]$*,* $\tau : V \to \mathbb{N}$*,* $Fc : N \times \mathbb{N} \to \mathbb{R}$*,*
- $subst : C \uplus N \to SubstList$ *(with* $SubstList$ *as in table 1).*

Through this definition, all elements discussed above are included: networks contain cognitive edges (labelled with expectabilities) and normative edges (labelled with normative force values); $cond$ returns their conditions, and $subst$ their substitution lists. Nodes are labelled with message templates through the $mesg$ mapping, so that "$\triangleright$" occurs only at the root node, and neighbouring nodes' labels never match (otherwise the expectability condition in equation 3 would be corrupted).

The only two elements of this definition that have not been discussed so far are the horizon constant H, which denotes the scope of maximal message sequence length for which the EN is relevant, and the total number of "non-continuations" n. Usually, this will be incremented each time a node (1) is appended to the root node, (2) matches one of the children nodes of the root node. Both are necessary for defining the semantics of the EN, which are discussed in detail in the following section.

2.5 Formal Semantics

For an arbitrary set S, let $\Delta(S)$ be the set of all (discrete) probability distributions over S with finite support. We define the semantics $I_{EN}(KB, w)$ in a network EN as a mapping from knowledge base states and current message sequence prefixes to the posterior probability distributions over all possible postfixes (conclusions) of the communication. Formally,

$$I_{EN}(KB, w) = f, \quad f \in \Delta(\mathcal{M}_c^*) \tag{5}$$

where

$$f(w') = \frac{g(w'\bot)}{\sum_{v \in \mathcal{M}_c^*} g(v\bot)} \tag{6}$$

is defined as the normalised value of $g(w'\bot)$, which represents the probability that w will be concluded by message sequence w', for any $w, w' \in \mathcal{M}^*$. We compute the probability for $w'\bot$ to make sure w' is followed by a node with label $\bot$ in the network, because the probability of w' is the probability with which the communication sequence will *end* after $w'_{|w'|}$ (and not that w' will simply be the prefix of some longer sequence). Also note that the sum in the denominator is not, as it may seem, infinite, because f has finite support.

Informally, the probability of w' should be inferred from multiplying all the expectability weights along the cognitive path that matches w' (if any), and increasing/decreasing these values with current force values from normative edges, if such edges end in nodes on this matching path. Before presenting the top-level formula for $g(w')$, we need some auxiliary definitions:

Firstly, we need to determine the node in a network EN that corresponds to a word w, which we denote by $mesg^{-1}$:

$$mesg^{-1}(\varepsilon) = v \quad :\Leftrightarrow \quad mesg(v) = \triangleright$$

$$mesg^{-1}(wm) = \begin{cases} v' & \text{if } \exists (v, v') \in C(KB(w)).\exists \vartheta \in subst(v, v'). \\ & (mesg(v') \cdot subst(w)\vartheta = m \wedge mesg^{-1}(w) = v) \\ \bot & \text{if no such } v' \text{ exists} \end{cases} \tag{7}$$

if $w \in \mathcal{M}_c^*$, $m \in \mathcal{M}_c$. The first case states that the node corresponding to the empty sequence ε is the unique root node of (V, C) labelled with $\triangleright$. According to the second case, we obtain the node v' that corresponds to a sequence wm if we take v' to be the successor of v (the node reached after w) whose label matches m under certain conditions:

- Edge (v, v') has to be a cognitive edge that is available in $EN(KB)$, where the elements of prefix w have already been executed. Since these can be physical actions, we must capture the restriction imposed on possible successors by having executed physical actions. Let $\xrightarrow{m}$ a function that modifies the knowledge base after message m is uttered. For a knowledge base KB, we can define

  $$KB(w) = KB' \;:\Leftrightarrow\; KB \xrightarrow{w_1} \ldots \xrightarrow{w_{|w|}} KB'$$

  so that the successors considered in each step for determining $mesg^{-1}(w)$ always take into account the consequences of previous actions[5]. Therefore, (v, v') has to be in the set of cognitive edges $C(KB(w))$ still feasible after w.
- There has to be a substitution $\vartheta \in subst(v, v')$ which, when composed with the substitution $subst(w)$ applied so far to obtain the messages in w_1 to $w_{|w|}$ from the respective nodes in EN, will yield m if applied to $mesg(v')$. This is expressed by $mesg(v') \cdot subst(w)\vartheta = m$. In other words, there is at least one combined (and non-contradictory) variable substitution that will make the node labels along the

[5] Note also that $mesg^{-1}(w)$ can only be determined unambiguously, if for any knowledge base content, a unique cognitive successor can be determined (e.g. by ensuring that equations 3 and 4 hold). This is another reason for the constraint regarding non-unifying out-links of nodes in definition 1.

path $mesg^{-1}(wm)$ yield wm if it is applied to them (concatenating substitutions is performed in a standard fashion). Thereby, the following inductive definition can be used to derive the substitution $subst(w)$ for an entire word w:

$$\begin{aligned} w = \varepsilon: &\quad subst(w) = \langle\rangle \\ w = w'm: &\quad subst(w) = subst(w') \cdot unifier(mesg(mesg^{-1}(wm)), m) \end{aligned}$$

where $\cdot$ is a concatenation operator for lists and $unifier(\cdot,\cdot)$ returns the most general unifier for two terms (in a standard fashion). Thus, $subst(w)$ can be obtained by recursively appending the unifying substitution of the message label of each node encountered on the path w to the overall substitution.

With all this, we are able to compute $g(w')$ as follows:

$$g(w') = \begin{cases} |\cup_{i=1}^{H} \mathcal{M}_c^i|^{-1} & \text{if } \exists v \in out(mesg^{-1}(w)).mesg(v) =? \\ \prod_i \left(\sum_{e \in pred(ww',i)} S(e)\right) & \text{else} \end{cases} \tag{8}$$

which distinguishes between two cases: if the path to node $mesg^{-1}(w)$ whose labels match w (and which is unique, because the labels of sibling nodes in the EN never unify) ends in a "?" label, the probability of a w' is simply one over the size of all words with length up to the horizon constant H (hence its name). This is because the semantics of "?" nodes is "don't know", so that all possible conclusions to w are uniformly distributed. Note that this case actually only occurs when new paths are generated and it is not known where they will lead, and also that if an outgoing link of a node points to a node with label "?", then this node will have no other outgoing links.

In the second case, i.e. if there is no "?" label on the path p from $mesg^{-1}(w)$ to $mesg^{-1}(ww')$, then the probability of w' is the product of weights $S(e)$ of all edges e on p. Thereby, $S(e)$ is just a generalised notation for expectability or normative force depending on the typed edge, i.e. $S(e) = Expect(e)$ if $e \in C$ and $S(e) = Fc(e, \tau(source(e)))$ if $e \in N$. The sum of these S-values is computed for all ingoing edges $pred(ww', i)$ of the node that represents the ith element of w', formally defined as

$$\forall w \in \mathcal{M}_c^*.pred(w,i) = \begin{cases} in(mesg^{-1}(w_1 \cdots w_i)) & \text{if } mesg^{-1}(w_1 \cdots w_i) \neq \bot \\ \emptyset \end{cases} \tag{9}$$

Note that summing over edges $pred(w)$ we calculate the sum of the (unique) *cognitive* predecessor of the node corresponding to $w_{|w|}$ and of all ingoing *normative* edges ending in that node. Finally, we compute the product of the probabilities along the w' path to obtain its overall probability. Looking back at the definition of $mesg^{-1}$, if no appropriate successor exists for m, the function returns $\bot$ (and $pred$ returns $\emptyset$, so that the probability $g(w')$ becomes 0 for continuations w' for which there is no path in the network). It is important to understand that condition

$$\begin{aligned} \text{if } \exists (v, v') \in C(KB(w)).\exists \vartheta \in subst(v, v') \\ (mesg(v') \cdot subst(w)\vartheta = m \wedge mesg^{-1}(w) = v) \end{aligned}$$

of equation 7 implies that only those continuations w' of a prefix w will have a non-zero probability that are *identical* to the result of substituting a message label by one of the existing substitutions. Using this definition, the generalisation aspect of the EN is quite weak, as it only allows for generating "lists" of concrete cases.

Of course, alternatives to this can be applied, e.g.

$$\ldots \exists \vartheta \in \mathit{SubstList} \ldots$$

which would allow *any* substitution to be applied to the node labels (and render edge substitution lists useless), or

$$\ldots \exists \vartheta \in \Big(\mathit{subst}(v, v') \cup \big(\mathit{SubstList} - \textstyle\bigcup_{e \neq (v,v'), e \in out(v)} \mathit{subst}(e) \big) \Big) \ldots \qquad (10)$$

which would allow any substitution that (i) either pertains to the substitution list of (v, v') or (ii) that makes *not* part of one of the substitution lists of outgoing links of v other than (v, v'). The intuition here is that "unless the substitution in question indicates following a different path, it may be applied". In fact, we will use condition 10 unless stated otherwise, because we assume a maximally general interpretation useful, which can of course be further restricted by semantic constraints in the edge conditions to yield arbitrary criteria for message pattern matching.

This concludes the definition of the semantics of a message (sequence) in a given expectation network. Essentially, all the formalisms introduced above allow for capturing statistical as well as normative knowledge about possible communication behaviour in a system in a compact fashion: each edge is tagged with logical constraints, and each potential path can be interpreted in an adjustably general sense by using appropriate variable substitution lists for the edges. Then, we defined the meaning of a message (sequence) as an estimate of its potential consequences in terms of *concrete* message sequences.

A final remark should be made about the use of performatives in this model. Their use should by no means imply that we expect them to have fixed semantics or induce reliable mentalistic properties on the parties involved. Much more, we employ them as "markers" for paths in the ENs, that can – unlike all other parts of the messages – *not* be replaced by variables. The intuition behind this is that there should be a non-generalisable part of each message that forces the observer to make a distinction. Next, we define communication systems as mathematical structures that operate on ENs.

3 Communication Systems

A communication system describes the evolutionary dynamics of an expectation network. The main purpose of a CS is to capture changes to the *generalised* meaning of communicative action sequences in the course of interaction in a multiagent system, in contrast to expectation networks themselves, which model meaning changes of certain messages in dependence of the message context (i.e., its preceding message sequences within the EN) only. These changes, which can be expressed in terms of expectations about future behaviour, are derived from statistical observation. However, they may be biased by the beliefs and belief transformation functions of the CS, i.e. the CS is an

autonomous observer that may have its own goals according to which it biases the expectations it computes. In contrast to agents who reason about expectations (such as InFFrA agents, cf. section 4), though, a CS need not necessarily be an active agent who takes action in the MAS itself, as described in section 1. Describing how communication systems work should involve (at least) clarifying:

- which communicative actions to select for inclusion in an EN,
- where to add them and with which expectability (in particular, when to consider them as "non-continuations" that follow "▷"),
- when to delete existing nodes and edges (e.g. to "forget" obsolete structures), and how to ensure integrity constraints regarding the remaining EN,
- when to spawn insertion/deletion of normative edges and with which normative force/content/condition/substitutions.

A formal framework for specifying the details of the above is given by the following, very general, definition:

Definition 2. *A* communication system *is a structure*

$$CS = (\mathcal{L}, \mathcal{M}, f, \kappa)$$

where

- *$\mathcal{L}$, $\mathcal{M}$ are the formal languages used for logical expressions and messages (cf. table 1),*
- *$f : \mathcal{EN}(\mathcal{L}, \mathcal{M}) \times \mathcal{M}_c \to \mathcal{EN}(\mathcal{L}, \mathcal{M})$ is the* expectation update function *that transforms any expectation network EN to a new network upon experience of a message $m \in \mathcal{M}_c$,*
- *$\kappa : 2^{\mathcal{L}} \times \mathcal{M}_c \to 2^{\mathcal{L}}$ is a* knowledge base update function *that transforms knowledge base contents after a message accordingly,*

and $\mathcal{EN}(\mathcal{L}, \mathcal{M})$ is the set of all possible expectation networks over $\mathcal{L}$ and $\mathcal{M}$.

The intuition is that a communication system can be characterised by how it would update a given knowledge base and an existing expectation network upon newly observed messages $m \in \mathcal{M}_c$. This definition is very general, as it does not prescribe how the EN is modified by the CS. However, some assumptions are reasonable to make (although not mandatory):

- If EN is converted by f via EN' to EN'' if m and m' are observed successively (with τ' and τ'' the respective times since the last observation of a label),

$$\tau''(v) = \begin{cases} 0 & \text{if } (\exists v' \in \mathit{pred}(v).\tau'(v') = 0 \wedge \mathit{unify}(m', \mathit{mesg}(v)) \\ 0 & \text{if } \mathit{pred}(v) = \{v_0\} \wedge \mathit{mesg}(v_0) = \triangleright \wedge \mathit{unify}(m', \mathit{mesg}(v)) \\ \tau'(v) + 1 & \text{else} \end{cases}$$

  So, the τ-value is reset to 0 if v is a successor of the root node, or if its predecessor's τ-value was just reset to 0 and the node in question matches the current message m'. Effectively, this means that the duration since the last occurrence of a node is incremented for all those nodes who have not occurred as successors of nodes that occurred in the previous step.

- If KB is the current knowledge base, $\kappa(KB, m) \models KB(m)$ should hold, so that all facts resulting form execution of m are consistent with the result of the κ-function.
- If any message sequence w' has occurred with frequency $\Pr(ww')$ as a continuation of w in the past, and EN_C is the same as EN reduced to cognitive edges, $I_{EN_C}(KB, w)(w') = \Pr(ww')$ should be the case, i.e. any bias toward certain message sequences not based on empirical observation should stem from normative edges. Note that this requirement says nothing about the probabilities of sequences never experienced before which result from applying the criterion in equation 7.

Normative edges left aside, an EN should predict the future of the respective observable communication sequences as accurately as possible. To achieve this, the respective CS is supposed to provide at least the following functionality:

Message Filtering and Syntax Recognition. Depending on its personal goals and the application domain, the observer might not be interested in all observable messages. Since ENs may not provide for *a priori* expectations, the discarding of such "uninteresting" messages can only take place *after* the semantics (i.e., the expected outcome) of the respective messages has already been derived from previous observation. Because discarding messages bears the risk that these messages become interesting afterwards, as a rule of thumb, message filtering should be reduced to a minimum. More particularly, messages should only be filtered out in cases of more or less settled expectations. Paying attention to every message and filtering uninteresting or obsolete information later by means of structure reweighting and filtering (cf. below) is presumably the more robust approach.

A very important feature of communication languages is their ability to effectively encode the generalised meaning of utterances by means of syntax. Our computationally tractable approach [13] to this phenomenon relies on the assumption that the syntax of messages somehow reflects expectation structures which have already been assembled.

Structure Expansion. Structure expansion is concerned with the growth of an EN in case a message sequence is observed which has no semantics defined by this EN yet. In theory, such an expansion would never be necessary, if we could initially generate a *complete EN* which contains dedicated paths for *all* possible message sequences. In this case, the observer would just have to keep track of the perceived messages and to identify this sequence within the EN to derive its semantics (provided there is no "semantic bias" in form of observer predispositions or norms).

For obvious reasons, such a complete EN cannot be constructed in practice. In contrast, the most general *minimal EN* would yield "?" for every message sequence, thus being of no practical use. As a solution, we could start with the minimal EN and incrementally add a node for each newly observed message. This is still not smart enough, because it does not take advantage of generalisable message sequences, i.e. different sequences that have the same approximate meaning. In general, such a generalisation requires a relation which comprises "similar" sequences. The properties of this relation of course depend on domain- and observer-specific factors. A simple way of generalising is to group messages which can be unified syntactically.

Garbage Collection. Several further methods of EN processing can be conceived of that aid in keeping the computation of (approximate) EN semantics tractable. This can be achieved by continuously modifying expectation structures using certain meta-rules, for example:

1. "fading out" obsolete observations by levelling their edge weights;
2. replacing (approximately) uniform continuation probability edges with "?";
3. removal of "?"s that are not leafs;
4. keeping the EN depth constant through removal of one old node for each new node;
5. removal of very unlikely paths.

4 Integrating the Agent Perspective

In this section, we will explain how the Interaction Frames and Framing Architecture InFFrA [16, 17] fits into the communication systems view. Interestingly enough, despite the fact that this architecture was developed independently from the CS framework using interactionist theories (yet also based on the socionic principles discussed in section 1), it soon proved to be very similar to it. As a result, we have tried to adapt the notation of both CS and InFFrA so as to minimise variations between them. The re-interpretation of framing-based systems as communication systems is useful, because it

- explains how InFFrA is a "special case" of CS, specifically designed for practical, agent-level social learning and decision-making;
- points at strengths and limitations of InFFrA that result directly from making specific design choices in the communication systems framework;
- is an example for agent-level use of CS (recall, though, that embedding CS in agents is just one possibility);
- illustrates how information can be exchanged between InFFrA agents and other system components that follow the communication systems paradigm by using ENs as an interface.

To achieve this re-interpretation, we will use the formal m²InFFrA model introduced in [16] (in this volume), based upon which we will discuss how CS might be mapped to InFFrA agents and vice versa. A brief overview of m²InFFrA shall suffice for this purpose. For the full m²InFFrA notation and definitions, please refer to [16].

4.1 Overview of m²InFFrA

m²InFFrA agents are agents that engage in discrete, turn-taking conversations (so-called encounters) 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 it strategically in future encounters. Any such frame is a structure $F = (T, C, \Theta, h)$ that consists of a trajectory T, lists of conditions/substitutions C/Θ and an occurrence counter vector (we write $T(F)$, $C(F)$, $\Theta(F)$ and $h(F)$ for the respective elements of frame F).

The meaning of such a frame is best explained by an example:

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

This frame states that five encounters started with a message matching $\texttt{propose}(A,B,X)$, three of them continued with $\texttt{accept}(B,A,X)$, and two were concluded by agent A performing physical action X (we 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). The remaining two observations might be due to encounter termination after the first message or were continued with a message that does not match $\texttt{accept}(B,A,X)$, and one encounter either finished after $\texttt{accept}(B,A,X)$ or continued with something different from $\texttt{do}(A,X)$. Also, the agent stored the two sets of conditions (and respective substitutions) under which this frame occurred (where the ith substitution applies for the ith condition).

m²InFFrA agents who use such frames are defined as structures

$$a = (\mathcal{L}, \mathcal{M}, \mathcal{E}, n, u, f, \kappa, \sigma)$$

with logical/message pattern languages $\mathcal{L}$, $\mathcal{M}$ (deliberately identical to the languages introduced in table 1), a set of encounter names $\mathcal{E}$, a count n of all encounters perceived so far, a utility function u, functions f and κ that transform frame repository and knowledge base when an encounter is perceived. Finally, they employ a similarity measure σ for message pattern sequences which they use to derive a probability distribution for potential message sequences given their similarities to those stored in interaction frames in the repository.

Such a probability distribution is called a *framing state* $[a](\mathcal{F}, KB, w) \in \Delta(\mathcal{M}_c^*)$ that maps any frame repository and knowledge base contents $\mathcal{F}$ and KB to a probability distribution over message sequences. For any two message sequences w and w', this distribution assigns a probability that an encounter which started with w will be concluded with w' (e.g. if $[a](\mathcal{F}, KB, w)(w') = 0.3$, then an encounter that started with w will be concluded by w' with a probability of 30%).

4.2 Communication Systems and m²InFFrA

At first glance, quite some similarities between CS and m²InFFrA become obvious. Most prominently, these are

1. the use of message patterns to generalise from concrete instances of messages and the recording of past cases in the form of variable substitutions;
2. the conditioning of message sequences with logical constraints to restrict the scope of certain expectations;

3. the evolutionary semantics of messages, updated with new observations;
4. the formalisation of a social reasoning component (CS/agent) as an "expectation transformer" (cf. functions f and κ in both definitions).

However, there are also distinct differences, which shall be made concrete by discussing the possibility of converting expectation networks to frame repositories and vice versa, the central question being whether an m²InFFrA agent can be built for an arbitrary CS and vice versa.

Converting Frames to Expectation Networks. Up to minor difficulties, this conversion is quite straightforward. We sketch the most important steps while leaving out certain formal details. Any frame can be transformed to a set of so-called "singular" frames with only one condition set and no substitutions. For a frame F, this is achieved by generating the set

$$F_s = \left\{ (T(F)\vartheta, C[i]\vartheta, \langle\rangle, \mathbf{1}) \;\middle|\; \vartheta = \Theta(F)[i], 1 \leq i \leq |\Theta(F)| \right\}$$

Thus, any frame in F_s covers exactly one of the cases previously represented by the substitutions in $\Theta(F)$ (and its trajectory is variable-free up to variables in logical content expressions). To build an EN incrementally from a repository $\mathcal{F}$ that consists of singular frames only, we proceed as follows.

1. Add a root node v_0 with label "$\triangleright$" to the new network EN. Let $i = 1$.
2. For each $F \in \mathcal{F}$:
   (a) If $T(F)$ does not end with "?", set $T(F) \leftarrow \triangleright T(F)\bot$, else $T(F) \leftarrow \triangleright T(F)$.
   (b) Set $c = \bigwedge_j c_j$ where $C(F) = \langle\{c_1, \ldots, c_m\}\rangle$.
   (c) Search EN for the longest path p whose node labels match a prefix of $T(F)$. Let v be the last node of p (potentially the root node of EN).
   (d) If $|p| \geq |T(F)| - 1$ (i.e. p matches the trajectory at least up to the last symbol), then:
      – Let c' the condition list of the last edge of p.
      – If p ends in a "?" label, erase its last node and edge.
   (e) Construct a new path p' consisting of nodes for the postfix of $T(F)$ that does not appear on p. Append p' to v. Let v' the first node of p', and e' the edge connecting v with v'.
   (f) Set $cond(f) = c \vee c'$ where f is the *last* edge on the new path p'.
   (g) Update $Expect(e') \leftarrow ((i-1)Expect(e') + 1)/i$ and $Expect(e) \leftarrow ((i-1)Expect(e) + 1)/i$ for other outgoing edges $out(v)$ of v.
   (h) Set $Expect(e) = 1$ for all other new edges on p'.
   (i) Increment i.

The idea behind this conversion is to construct a single path for each distinct frame, where shared prefixes have to be merged. Each singular frame covers a single case that has occurred exactly once, and whose trajectory contains no variables (for which reason it has no substitution). Two cases can occur: either the frame ends in "?" or not. We prepend a $\triangleright$ to the trajectory, and append a $\bot$ symbol if "?" is not the last symbol (step 2a). Then, we look for the longest prefix of $T(F)$ that is already in the network (step 2c)

and append new nodes and edges for the remaining postfix. If (case 2d) the trajectory is already contained but possibly with previous "don't know" ending, we delete the last edge and node (step 2d) and memorise its condition c', so that we can add it in a disjunctive fashion to c in step 2f. Thus, if F itself has "don't know" semantics, two "?" nodes become one, and if it ends with $\perp$, the previous "don't know" semantics are not valid anymore. Also, in step 2f the new condition is "moved" to the very last edge of the new path. Expectability update (step 2g) is a matter of straightforward counting.

Thus, we obtain a very simple EN without normative edges and substitution lists, where all conditions (which summarise the frames with identical trajectories) only occur at the very last edge of any path (leading to $\perp$ or "?")[6].

Of course, it is not possible to prove that a CS can be constructed using this EN whose continuation estimates will be identical to the agent state of *any* m²InFFrA agent, especially because agents might apply arbitrary similarity measures σ. This is because there is no equivalent notion for this measure in CS (equation 7 is used instead to estimate the probability of new cases). However, if a very simple σ is used, which assigns similarity 1 to identical message patterns and 0 to all other comparisons, the construction of an equivalent CS is straightforward. The CS would simply generate a new singular frame for any encounter, and call the procedure sketched above after each newly observed message. Whenever the m²InFFrA agent starts a new encounter, this would be considered a non-continuation in the CS sense. This CS would compute the same probabilities as the original agent if the conditions in frames fulfil the conditions 3 and 4.

Converting Expectation Networks to Frames. In an attempt to convert ENs to frames, we might proceed in a similar manner by trying to generate a frame for every distinct path. This would imply

- substituting sender/receiver variables by new variables so that a turn-taking, two-party trajectory is obtained; this involves adding extra conditions that state which agent may hold which of these two roles in which step;
- joining all edge conditions on the path to obtain frame condition lists; however, conditions in ENs need only hold after the edge on which they appear, so a notion of time has to be introduced;
- generating a list with all possible substitutions occurring on the path to be used as a frame substitution list.

For lack of space, we cannot introduce all of these steps formally. Let us look at an example of a frame generated from path **P6** in figure 1 called F_6 shown in table 2: Variables A and B have been replaced by I(nitiator) and R(esponder), and the reverse substitutions have been pre-pended to the substitutions in Θ. Although this is not necessary in this frame, frames for paths **P2** or **P3** would require introduction of these new variables, as they involve messages subsequently sent by the same agent (**P2**) or more than two parties (**P3**). Also, by adding the statements $current(E)$ and

[6] For EN semantics, the edge on which conditions occur on a path do not matter, but, of course, this EN would be very inefficient for computing continuation probabilities because all paths have to be followed to the end before their logical validity can be checked.

$$F_6 = \Big\langle \langle \overset{100}{\rightarrow} \texttt{request}(I,R,X) \overset{50}{\rightarrow} \texttt{propose}(R,I,Y) \overset{25}{\rightarrow} \texttt{accept-proposal}(I,R,Y) \rangle,$$
$$\langle \{current(E), message(M,E,1) \Rightarrow price > 0\},$$
$$\langle \{current(E), message(M,E,1) \Rightarrow price > 0\} \rangle,$$
$$\langle \langle [I/A], [R/B], [X/\texttt{deliver\_goods}], [Y/\texttt{pay\_price}], [A/\texttt{agent\_1}], [B/\texttt{agent\_2}] \rangle,$$
$$\langle [I/A], [R/B], [X/\texttt{deliver\_goods}], [Y/\texttt{pay\_price}], [A/\texttt{agent\_1}], [B/\texttt{agent\_3}] \rangle \rangle \Big\rangle$$

Table 2. A frame for path **P6**.

$message(M,E,1)$ as preconditions to $price > 0$, where $current(E)$ means that E is the current encounter, $price > 0$ need only hold after the first message, as in the EN. This "contextualisation" of conditions has to be performed for each original edge condition. Thirdly, we need to generate all feasible substitution list combinations along all edges, as is shown by the substitution lists in F_6 which contain both cases $[A/\texttt{agent\_1}], [B/\texttt{agent\_2}]$ and $[A/\texttt{agent\_1}], [B/\texttt{agent\_3}]$. However, a problem appears here, which is that we cannot discriminate whether $[A/\texttt{agent\_1}], [B/\texttt{agent\_3}]$ is an actual case of **P6**: looking at **P2**, we can see that $[C/\texttt{agent\_3}]$ contradicts $[B/\texttt{agent\_3}]$, so it seems logical that $[A/\texttt{agent\_1}], [B/\texttt{agent\_3}]$ yields **P6**. But which conclusion of the encounter does $[A/\texttt{agent\_1}], [B/\texttt{agent\_2}]$ belong to? We simply cannot tell.

There are several further reasons for which ENs cannot be converted into m²InFFrA frames by a generic procedure that will ensure equivalence of semantics:

1. Normative links may link non-subsequent messages statistically with each other. Such links exceed the expressiveness of frame trajectories, and although there may be ways to treat certain normative links by meta-rules in the agent's knowledge base, there is no generic procedure of generating frames with these conditions, because the effects of normative links are *non-local* to frames.
2. Cognitive links may link messages that do not represent continuations occurring *within* encounters. These cannot be included in frame trajectories, because trajectories end whenever encounters end.
3. Even if we know the total number of non-continuations n, no frame counters can be reconstructed for edges with different conditions. For example, in figure 1, the distribution of outgoing edges of $\texttt{request}(A,B,X)$ between cases $price = 0$ and $price > 0$ is not available, so that some hypothetical (say 50/50) distribution between would have to be made up.
4. The computation of continuation probabilities in ENs proceeds by "identification with previous cases and exclusion of cases on different paths" as reflected by condition 7. For example, after $\texttt{request}(\texttt{agent\_3}, \texttt{agent\_2}, X)$, the possibility of $\texttt{propose}(\texttt{agent\_3}, \texttt{agent\_2}, X)$ (continuation with **P3**–**P7**) is ruled out by $[A/\texttt{agent\_3}], [B/\texttt{agent\_2}]$ appearing on **P8**. There is no way a similarity measure σ can be conceived that can reflect this solely by comparing

$[A/\texttt{agent\_3}]$, $[B/\texttt{agent\_2}]$ to the previous cases $[A/\texttt{agent\_1}]$, $[B/\texttt{agent\_2}]$ and $[A/\texttt{agent\_1}]$, $[B/\texttt{agent\_3}]$ without any non-local information.

All these nicely illustrates what we would expect of a sociological comparison between systems theory and interactionism, namely that the general theory of communication systems obviously subsumes interactionist approaches, since interaction systems are specific kinds of communication systems tied to a number of particular assumptions. These are: (i) co-presence and spatial/temporal proximity of context-generating communication, (ii) ego/alter distinctions, (iii) locality of processing of social knowledge ("blindness" to expectations relevant in other contexts during involvement in a particular encounter), (iv) expectation formation by analogy (rather than possibility) and (v) simplicity in the representation of expectations (to ensure efficient processing with bounded cognitive resources).

The fact that these elements have been made concrete and that distinctions between several socio-theoretical approaches have been mapped to formal computational models constitutes a major advance in Socionics as it not only furthers the understanding of the underlying theories, but is also instructive in the identification of how to apply CS and frame-based agents in different applications. Some of these potential applications will be discussed in the following section.

5 Discussion: Applications and Extensions

The modelling of social structures on the basis of expectation networks and communication systems allows for novel approaches to a variety of DAI themes. We review (i) identification of ontologies for inter-agent communication and – closely related – the finding of verifiable and flexible semantics for agent communication languages; (ii) *mirror holons* as a new model for holonic theories of agency and software engineering methods based on expectation-oriented modelling and analysis of multiagent systems.

5.1 Social Ontologies

In DAI, an ontology is a set of definitions as a means to provide a common ground in the conceptual description of a domain for communication purposes. Ontologies are usually represented as graphical hierarchies or networks of concepts, topics or classes, and either top-down imposed on the agents or generated in a bottom-up fashion by means of ontology negotiation. In a similar way, expectation networks are descriptions of the social world in which the agents exist. But ENs do not only describe social (i.e. communication) structures, but indirectly also the communication-external environment the message content informs about. Thus, communication systems can be used, in principle, for an incremental collection of ontological descriptions from different autonomous sources, resulting in stochastically weighted, possibly conflicting, competitive and revisable propositions about environmental objects. The crucial difference to traditional mechanisms is that such a *social ontology* represents expectations about how a certain object will be described in future communication. This opposes the "imposed ontologies" view somewhat, where the ontology provides an *a priori* grounding for communication, and makes this approach appear particularly suitable for open multiagent

systems with a highly dynamic environment, where homogenous perception among agents cannot be assumed. Also, it is appropriate whenever descriptions are influenced by individual preferences such that a consensus cannot be achieved (think, e.g., about "politically" biased resource descriptions in the context of the Semantic Web [12]). In the following, we will sketch two approaches for extracting social ontologies from expectation networks.

Extraction of Speech Act Types. The current version of FIPA-ACL [4] provides an extensible set of speech-act performative types with semantics defined in a mentalistic fashion. In our approach, we can imagine some system component (e.g., a so-called multiagent system *mirror* [13, 8]) that provides the agents with a set of performatives *without* any predefined semantics and wait for the semantics of such "blank" performatives to emerge. To become predictable, it is rational for an agent to stick to the meaning (i.e., the consequences) of performatives, at least to a certain extent. This meaning has been previously (more or less arbitrarily) "suggested" for a certain performative by some agent performing demonstrative actions after uttering it.

Of course, a single agent is not able to define a precise and stable public meaning for these performatives, but at least the intentional attitude associated with the respective performative needs to become common ground to facilitate non-nonsensical, non-entropic communication [15]. A particular performative usually appears at multiple nodes within the EN, with different consequences at each position, depending on context (especially on the preceding path), message content and involved sender and receiver. To build up an ontology consisting of performative types, we have to continually identify and combine the occurrences of a certain performative within the current EN to obtain a general meaning for this performative (i.e. a "type" meaning). Afterwards, we can communicate this meaning to all agents using some technical facility within the multiagent system, like a middle agent, a MAS mirror or an "ACL semantics server". Of course, such a facility cannot impose meaning in a normative way as the agents are still free to use or ignore public meaning as they like, but it can help to spread language data like a dictionary or a grammar does for natural languages. The criteria for the identification and extraction of performative meaning from ENs are basically the same as the criteria we proposed in section 3 for the generalisation over message sequences.

Extraction of Domain Descriptions. While a set of emergent speech act types constitutes a social ontology for communication events, classical ontologies provide a description of an application domain. To obtain a social version of this sort of ontology from an EN, two different approaches appear to be reasonable: (1) Inclusion of environment events within the EN and (2) probabilistic weighting of assertions.

The former approach treats "physical" events basically as utterances. Similar to the communicative reflection of agent actions by means of `do`, a special performative `happen`$(event)$ would allow EN nodes that reflect events occurring in the environment. These events will be put in the EN either by a special CS which is able to perceive the agents' common environment, or by the agents themselves as a communicative reflection of their own perceptions. A subset of $event$ is assumed to denote events with consensual semantics (think of physical laws), i.e., the agents are not free to perform

any course of action after such an event has occurred, whereas the remainder of $event$ consists of event tags with open semantics that has to be derived empirically from communications observation just as for "normal" utterances. If such an event appears the first time, the CS does not know its meaning in terms of its consequences. Its meaning has thus to be derived *a posteriori* from the communicational reflection of how the agents react to its occurrence. In contrast, approach (2), which we proposed for the agent-based competitive rating of web resources [12], exploits the propositional attitude of utterances. The idea is to interpret certain terms within $LogicalExpr$ as domain descriptions and to weight these descriptions according to the amount of consent/dissent (using predefined performatives like $Assert$ and $Deny$). The weighted propositions are collected within a knowledge base (e.g., KB) and are communicated to the agents in the same way as the emergent speech act types before. Unlike approach (1), ontologies are constructed "by description" not "by doing" in this way. The advantage of approach (1) lies in its seamless integration of "physical" events into the EN, whereas (2) is probably more easy to apply in practice.

5.2 Mirror Holons: Multi-Stage Observation and Reflection

In [1, 8], we have introduced the *social system mirror* architecture for open MAS. The main component of this architecture is a so-called social system mirror (or "mirror" for short), a middle agent which continually observes communications, empirically derives emergent expectation structures (represented as an expectation network, which might also contain normative structures) from these observations, and "reflects" these structures back to the agents. Its goals are to influence agent behaviour by means of system-wide propagation of social structures and norms to achieve quicker structure evolution and higher coherence of social structures without restricting agent autonomy, and the provision of a representation of a dynamic communication system for the MAS designer. While a mirror only models a single communication system, and, except for the propagation of expectations, is a purely passive observer, the successor architecture *HoloMAS* [13] is able to model multiple communication systems at the same time through multiple *mirror holons* in order to model large, heterogenous systems. In addition, a mirror holon can take action itself by means of the execution of social programs which are generated from emergent expectation structures. "Ordinary agents" (and other mirror holons) can optionally be involved in this execution process as *effectors*, which realise holon commands within their physical or virtual application domain (unless they deny the respective command). In any case they can influence the social programs within a mirror holon through the irritation of expectation structures by means of communication. A mirror holon thus represents and (at least to some extent) replaces the functionality of the ordinary agents that contribute to the emergence of the respective expectation structures, but it does not disregard the autonomy of his adjoint actors. Another difference between mirror holons and traditional agent holons [18] is that a mirror holon does not represent or contain groups of agents, but instead a certain functionality which is identified in form of regularities in the observed communications. This functionality is extracted and continually adopted from dynamic expectation structures regarding criteria like consistency, coherence and stability, corresponding to the criteria sociological systems theory ascribes to social programs [9]. Mirror holons pave

the way for applications in which agent autonomy should not (or cannot) be restricted on the one hand, while reliable, time-critical system behaviour is desired. They can also be used as representants for entire communication systems (e.g., virtual organisations) that behave smoothly towards third parties whenever the communication system itself lacks coherence due to, for example, inner conflicts.

Expectation-Oriented Software Development. It has been long recognised that due to new requirements arising from the complex and distributed nature of modern software systems the modularity and flexibility provided by object orientation is often inadequate and that there is a need for encapsulation of robust functionality at the level of software components. Agent-oriented approaches are expected to offer interesting prospectives in this respect, because they introduce interaction and autonomy as the primary abstractions the developer deals with.

However, although interaction among autonomous agents offers great flexibility, it also brings with it contingencies in behaviour. In the most general case, neither peer agents nor designer can "read the mind" of an autonomous agent, let alone change it. While the usual strategy to cope with this problem is to restrict oneself to closed systems, this means loosing the power of autonomous decentralised control in favour of a top-down imposition of social regulation to ensure predictable behaviour. The EXPAND method (Expectation-oriented Analysis and Design) [1] follows a different approach. EXPAND is based on expectation networks as a primary modelling abstraction which both system designer and agents use to manage the social level of their activities. This novel abstraction level is made available to them through a special version of the social system mirror very similar to a CASE tool. For the designer, this mirror acts as an interface he uses to propagate his desired expectations regarding agent interaction to the agents and as a means for monitoring runtime agent activity and deviance from expected behaviour. For agents, this mirror represents a valuable "system resource" they can use to reduce contingency about each other's behaviour. EXPAND also describes an evolutionary process for MAS development which consists of multiples cycles: the modelling of the system level, the derivation of appropriate expectation structures, the monitoring of expectation structure evolution and the refinement of expectation structures given the observations made in the system. For a lack of space, we have to refer the interested reader to [1] for details.

6 Conclusion

This paper introduced communication systems as a unified model for socially intelligent systems based on recording and transforming communicative expectations. We presented formalisms for describing expectations in terms of expectation networks, the formal semantics of these networks, and a general framework for transforming them with incoming observation. Then, we exemplified the generic character of CS by analysing its relationship to micro-level social reasoning architecture using the InFFrA architecture as an example. Finally, a number of interesting applications of CS were discussed, some of which have already been addressed by our past research, while others are currently being worked on.

While a lot of work still lies ahead, we strongly believe that, by virtue of their general character, CS have the potential of becoming a unified model for speaking about methods and applications relevant to Socionics. Also, we hope that they can contribute to bringing key insights of socionic research to the attention of the mainstream DAI audience, as they put emphasis on certain aspects of MAS that are often neglected in non-socionic approaches.

References

1. W. Brauer, M. Nickles, M. Rovatsos, G. Weiß, and K. F. Lorentzen. Expectation-Oriented Analysis and Design. In *Procs. AOSE-2001*, LNAI 2222, Springer-Verlag, Berlin, 2001.
2. P. R. Cohen and H. J. Levesque. Performatives in a Rationally Based Speech Act Theory. In *Procs. 28th Annual Meeting of the ACL*, 1990.
3. P. R. Cohen and H. J. Levesque. Communicative actions for artificial agents. In *Procs. ICMAS-95*, 1995.
4. FIPA, Foundation for Intelligent Agents, http://www.fipa.org.
5. E. Goffman. *Frame Analysis: An Essay on the Organisation of Experience*. Harper and Row, New York, NY, 1974.
6. F. Guerin and J. Pitt. Denotational Semantics for Agent Communication Languages. In *Procs. Agents'01*, ACM Press, 2001.
7. Y. Labrou and T. Finin. Semantics and conversations for an agent communication language. In *Procs. IJCAI-97*, 1997.
8. K. F. Lorentzen and M. Nickles. Ordnung aus Chaos – Prolegomena zu einer Luhmann'schen Modellierung deentropisierender Strukturbildung in Multiagentensystemen. In T. Kron, editor, *Luhmann modelliert. Ansätze zur Simulation von Kommunikationssystemen.* Leske & Budrich, 2002.
9. N. Luhmann. *Social Systems*. Stanford University Press, Palo Alto, CA, 1995.
10. 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.
11. G. H. Mead. *Mind, Self, and Society*. University of Chicago Press, Chicago, IL, 1934.
12. M. Nickles. Multiagent Systems for the Social Competition Among Website Ratings. In *Procs. ACM SIGIR Workshop on Recommender Systems*, 2001.
13. M. Nickles et al. Multiagent Systems without Agents – Mirror-Holons for the Derivation and Enactment of Functional Communication Structures. In this volume.
14. J. Pitt and A. Mamdani. A protocol-based semantics for an agent communication language. In *Procs. IJCAI-99*, 1999.
15. M. Rovatsos, M. Nickles, and G. Weiß. Interaction is Meaning: A New Model for Communication in Open Systems. In *Procs. AAMAS'03*, Melbourne, Australia, to appear, 2003.
16. M. Rovatsos and K. Paetow. On the Organisation of Social Experience: Scaling up Social Cognition. In this volume.
17. M. Rovatsos, G. Weiß, and M. Wolf. An Approach to the Analysis and Design of Multiagent Systems based on Interaction Frames. In *Procs. AAMAS'02*, Bologna, Italy, 2002.
18. M. Schillo and D. Spresny. Organization: The Central Concept of Qualitative and Quantitative Scalability. In this volume.
19. M. P. Singh. A semantics for speech acts. *Annals of Mathematics and Artificial Intelligence*, 8(1–2):47–71, 1993.
20. M. P. Singh. A social semantics for agent communication languages. In *Procs. IJCAI Workshop on Agent Communication Languages*, 2000.