KNOWLEDGE-BASED MODELING AND NATURAL COMPUTING FOR COORDINATION IN PERVASIVE ENVIRONMENTS

Michael Cebulla
Technische Universität Berlin, Institut für Softwaretechnik und Theoretische Informatik,
Franklinstraße 28/29, D-10587 Berlin,

Keywords: Knowledge Engineering, Coordination in Multi-Agent Systems, Systems Engineering Methodologies.

Abstract: In this paper we start with the assumption that coordination in complex systems can be understood in terms of presence and location of information. We propose a modeling framework which supports an integrated view of these two aspects of coordination (which we call *knowledge diffusion*). For this sake we employ methods from ontological modeling, modal logics, fuzzy logic and membrane computing. We demonstrate how these techniques can be combined in order to support the reasoning about knowledge and very abstract behavioral descriptions. As an example we discuss the notion of distributed action and show how it can be treated in our framework. Finally we exploit the special features of our architecture in order to integrate bio-inspired coordination mechanisms which rely on the exchange of *molecules* (i.e. uninterpreted messages).

1 INTRODUCTION

Enabled by the recent advances on the fields of hardware design, wireless communication and (last not least) the Internet the distribution of mobile devices in society will dramatically increase during the forthcoming decade. In these pervasive environments especially the requirement of context-awareness is critical: mobile devices have to be aware of information and services which are present in the current situation and cooperate with them when desirable. On the other hand they are also required to provide meaningful behavior when these services cannot be found. Consequently pervasive systems have to possess novel interaction capabilities in order to be *aware* of the current situation and to initiate adequate behaviors. These capabilities are key features for the success of emerging pervasive technologies and are discussed under the topic *autonomous computing* (Kephart and Chess, 2003).

In this presentation we focus on the systems' ability to initiate cooperations with other systems and to represent knowledge about themselves and their situation. As we will see a special focus lies on issues related to the locality of knowledge and on the migration of information. We treat the first aspect of knowledge representation by relying on ontological modeling and knowledge bases thus enabling the high-level modeling of complex systems (as already proposed by (Pepper et al., 2002)). Regarding the second aspect related to the migration of knowledge we rely on *membrane computing* (Păun, 2000). This model which was taken from biology is highly appropriate to model systems which have to interact extensively with their environment. Systems which are modeled by membranes have porous system borders where molecules can intrude and exit. As we will see these molecules can be used to represent information which diffuses through the system.

For the syntactic representation of knowledge we propose to use ontologies (i.e. description logics (Baader et al., 2003)) as a light-weight formalism which provides support for automated reasoning. From our point of view such a formalism meets some requirements which may be unfamiliar from a traditional viewpoint concerning system modeling.

Intelligibility. Since it is very close to natural language and thus supports the direct usage of domain-specific terminology, ontological modeling provides an instrument for a *seamless knowledge management.*

Uncertainty and Incompleteness. Systems in adverse environments are subject to unexpected influences *(uncertainty)* and are characterized by a high complexity which makes an exact description impossible or inefficient. As we will see the issues of

Cebulla M. (2006).
KNOWLEDGE-BASED MODELING AND NATURAL COMPUTING FOR COORDINATION IN PERVASIVE ENVIRONMENTS.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - AIDSS*, pages 99-106
DOI: 10.5220/0002454500990106

vagueness and uncertainty are treated by the introduction of fuzzy logics (Klir and Yuan, 1995) and modal logics (Wooldridge, 1992) into terminological reasoning.

Highly Reactive Behavior. Since the semantics of membrane computing is based on multisets it is well-suited for the treatment of highly reactive behavior since e.g. inadequate assumptions concerning sequentiality are avoided.

Efficient Automated Reasoning. We claim that an enhanced intelligibility and support for incomplete specifications have to go hand in hand with well-defined semantics and efficient decision procedures. A standard way of reasoning is provided procedures which are based on *tableau algorithms* (Baader et al., 2003) which support a smooth integration of different aspects of knowledge. A key issue in this context is the adequate treatment of *implicit information*, e.g. with respect to structural similarities of different entities. Such a treatment is possible by exploiting the concept of *subsumption* as defined in the context of description logics.

This paper is organized as follows: first we give an outline of the general architecture (cf. Section 2). In this section we discuss the significance of observation-based behavioral modeling (cf. Subsection 2.1) and briefly introduce the basic concepts from membrane computing (cf. Subsection 2.2). Then we briefly discuss the basic concepts of fuzzy description logics (cf. Subsection 2.3). Section 3–4 give an overview about terminological reasoning concerning exemplary system aspects (e.g. architecture and behavior). We then show how to treat abstract specifications of distributed actions in our framework (cf. Section 5). Sections 4 and 5 can be considered as an exemplary discussion of tableau-based reasoning in the medium of membrane computing. In Section 6 we finally give an example for the integration of bio-inspired coordination supported by our framework. Throughout the paper we use examples from disaster management because we feel that the challenges related to context-awareness are very specific in these area.

2 GENERAL ARCHITECTURE

As we already indicated we doubt that traditional interaction mechanisms (like procedure calls or message passing) are qualified to support context-aware behavior in pervasive environments. For this reason we employ methods from the field of *natural computing* which are better suited to support the analysis of *knowledge diffusion.* Since we refer to mechanisms which were originally observed in biological or social

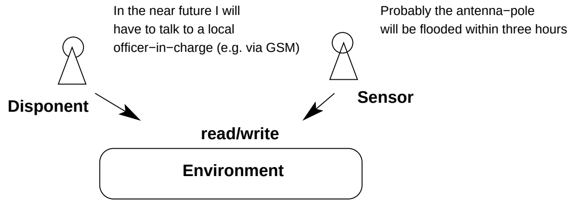

Figure 1: Typical Coordination Problem.

environments we start with examples from so-called socio-technical systems (e.g. coordination in disaster management). In this section we discuss a motivating example and develop the basic architecture.

2.1 Observation-based Modeling

We select *disaster management* as one of our target domains because there in the typical case the situation is characterized by high environmental dynamics, high need of information and frequent coordination problems. Obviously in this scenario (cf. Figure 1) the *water-level* is a highly important information (since it is one of the main parameters of the flooding). Thus it surely makes sense to observe this parameter and keep track of its changes. This is commonly done by *sensors* (which are represented by an unique sensor in our simplified architecture). Sensors collect information which then can be sent to the parts of the system where decisions are made on the basis of this information (e.g. the disponent). So far an initial model. But the real situation is far more complex. In fact the behavior of the flood (and thus the parameter *water-level*) cannot be considered as an isolated phenomena which can be observed by an independent observer. On the contrary the evolution of the situation (the flooding) directly affects the internals of systemic communication and decision making. Thus the rising of a flood has frequently severe consequences for the infrastructure (e.g. telephone nets for conventional or mobile communication). This dependency may lead to complex and hidden interactions between systemic aspects which seem to be unrelated at first sight. For instance the rising of the flood interacts with the disponent's need to use mobile telecommunication in the near future when an antenna pole will not be operable due to overflooding. Note that such hidden interactions between components are known as *complex interactions* (Leveson, 1995; Perrow, 1984) and are considered as major causes for losses and accidents.

Agents Knowledge and Behavior. We consider our approach as an extension of the modeling of multi-agent systems (cf. for example (Fagin et al., 1996)). For reasoning about knowledge and behavior

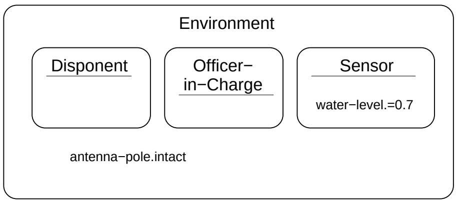

Figure 2: Scenario from Disaster Management (simplified).

we presuppose that each agent supports the following interface.

interface Agent
know: $X \times T \to \mathbf{bool}$
next: $X \times T \to X$

Such a signature is especially suited for the modeling of black box-behavior. While the function *know* supports *observations* concerning the current state of an agent the function *next* supports experiments concerning the behavior of an agent. Note that these functions have some coalgebraic flavour since the state space X of an agent may not be known completely (cf. (Rutten, 1996)). Thus these functions support the reasoning about an agent's state (his knowledge) and his behavior without requiring information about his internals.

While the *agent*-interface supports the observation of agents their *real* behaviors are described with concepts from membrane computing (cf. Subsection 2.2). Consequently, each agent is considered to be enclosed by a membrane. As we will see, due to this fact information can diffuse through systems and agents borders according to diffusion rules. We use concepts from multiset rewriting in order to describe the diffusion of knowledge through complex systems.

In Figure 2 we show some relevant agents in an highly simplified scenario of coordination described by our example. We show how to model agents as locations of information. We consider the knowledge which is situated *inside* the membranes as private to the agents while the information which is *outside* is accessible to all agents which share a certain environment.

Discussion. We can make several observation using this example. Firstly we claim that problems of this type can be modeled using the paradigm of knowledge diffusion. An important notion in this context concerns the location of information and question whether this location is suitable for this information or if it has to be moved to other locations. In addition we have to provide advanced coordination mechanisms which are highly robust in order to be able to process for example the tight connection between environment and communication. Consequently we propose to use the paradigm of knowledge diffusion not only for the analysis of coordination problems but also as a platform for very robust coordination. The related processes of knowledge diffusion are very similar to processes of biological knowledge processing (e.g. the usage of pheromones in insect populations, cf. (Krasnogor et al., 2005)).

Environment. Obviously in our approach the environment plays a prominent role. For illustration we distinguish two characteristic scenarios.

Naive Scenario. In the naive scenario the environment takes an adversial role. It does not provide knowledge about the situation but disturbs the agents' actions deliberately. The relevant knowledge is located exclusively in the agents. The weakness of this model consist in the fact that this knowledge does not suffice in complex and dynamic setting. Thus the agents have to specify interactions on the basis of incomplete knowledge. For example this will lead him to an inadequate selection concerning the communication channel.

Intelligent Environment. In the second case the environment incorporates relevant situational knowledge which can be used in order to enhance the systems behavior. The agents possess only incomplete knowledge about the situation. Consequently most of their actions have to be considered as interactions with the environment. The agents initiate only very abstract interactions while the environment is in charge to introduce the missing information. We consider this interaction with assistance by an intelligent environment as a typical case of *knowledge diffusion.* In this scenario agents do not address certain services but just give an expression of their needs to the environment.

The naive scenario perfectly describes the traditional view on systems which are not context-aware. In these systems the relevant knowledge is situated in the agents while the environment is considered to be *adversial.* In the case of intelligent environments on the other hand each action of an agent has to be considered as a *joint action* (also called distributed action) which is performed in cooperation with the environment.

2.2 Membrane Computing

More formally we represent knowledge as molecules (or terms) which are contained in multisets. The diffusion of knowledge can thus be described by multiset rewriting. For example, the diffusion of information in systems can described by simple rules:

$$[\mathsf{water\text{-}level.High}] \to [(\mathsf{need\text{-}to\text{-}act}, \mathsf{out})]$$

In this notation we use the brackets [and] for the specification of the membrane structure. In our example the brackets delimit the multiset which represents the knowledge which is private to the sensor. The resulting rule describes the distribution of information through the system. We deliberately use inexact terms like *need-to-act* in order to demonstrate the possibilities of high-level modeling. By the standard action out we specify that this molecule has to exit the sensor's local state. Note that this molecule is not expected to carry extensive semantic significance but may only transmit the information that the system's current state is not optimal.

P-Systems. Following (Păun, 2000) we use the concept of P systems which heavily relies on the metaphor of a *chemical solution* (Berry and Boudol, 1992) for the representation of knowledge in a system. As we already saw a solution contains *molecules* which represent *terms.* As we will see these terms are elements of a language which is describe by an ontology. More formally a P-System can be defined as follows (slightly adapted from (Păun, 2000)):

Definition 1 (P-System) *A P-system of degree m is defined as a tuple* $\Pi = \langle O, \mu, w_1, \ldots, w_m, R_1, \ldots, R_m \rangle$*, where* O *is an ontology,* μ *is a membrane structure,* $w_1, \ldots, w_m$ *are multisets of strings from* O *(representing the knowledge contained in regions* $1, 2, \ldots, m$ *of* μ*,* $R_1, \ldots, R_n$ *are sets of transformation rules associated with the regions.*

We chose this highly reactive semantic model as the basis of our process description because we feel that it is highly appropriate for the description of unexpected behavior. Especially, environmental changes or unexpected contextual influences can be modeled by introducing new molecules into the solution. In addition no assumptions concerning artificial sequentiality are imposed on the events contained in a multiset: in general case they can occur in every possible order.

2.3 Fuzzy Description Logics

For the representation of terms which may be subject to semantic interpretation we use fuzzy description logics. Following (Straccia, 2001) we introduce semantic uncertainty by introducing multi-valued semantics into description logics. Consequently we have to introduce fuzzy sets (Zadeh, 1965) instead of the crisp sets used in the traditional semantics (cf. (Baader et al., 2003)). For this sake we conceive the model of the terminological knowledge which is contained in a knowledge base as fuzzy set. When used in assertional statements we can express the fact that different instances (elements of Δ) may be *models* of a concept to a certain degree.

Definition 2 (Fuzzy Interpretation) *A* fuzzy interpretation *is a pair* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$*, where* $\Delta^{\mathcal{I}}$ *is, as for the crisp case, the domain whereas* $\cdot^{\mathcal{I}}$ *is an interpretation function mapping*

1. *individuals as for the crisp case, i.e.* $a^{\mathcal{I}} \neq b^{\mathcal{I}}$*, if* $a \neq b$*;*
2. *a concept* C *into a membership function* $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$*;*
3. *a role* R *into a membership function* $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$.

If C is a concept then $C^{\mathcal{I}}$ will be interpreted as the *membership degree function* of the fuzzy concept C w.r.t. $\mathcal{I}$. Thus if $d \in \Delta^{\mathcal{I}}$ is an object of the domain $\Delta^{\mathcal{I}}$ then $C^{\mathcal{I}}(d)$ gives us the degree of being the object d an element of the fuzzy concept C under the interpretation $\mathcal{I}$ (Straccia, 2001).

In this article we make use of the usual operators from description logics and silently introduce predicates on fuzzy concrete domains (which are very similar to linguistic variables and support the integration of linguistic hedges (Klir and Yuan, 1995)). We also heavily rely on the concept of *fuzzy subsumption* which we introduce by example. In addition we introduce *complex role*-terms in Section 5. Unfortunately we are not able to give an extensive treatment due to space limitations.

Fuzzy Subsumption. Intuitively a concept is subsumed by another concept (in the crisp case) when every instance of the first concept is also an instance of the second. In the fuzzy case, however, we are interested in the degree to which the current situation conforms to a certain concept. As an example we consider a case from the domain of disaster management. In the following we are interested in the degree to which a *current-situation* can be considered as a *flooding.*

$$\begin{aligned} \mathsf{flooding} &\doteq \mathsf{sit} \sqcap \exists \mathsf{water\text{-}level.very(High)} \\ \mathsf{curr\text{-}sit} &\doteq \mathsf{sit} \sqcap \exists \mathsf{water\text{-}level.} =_7 \end{aligned}$$

On this background we can reason about the following statement:

$$KB \quad \approx_{deg} \quad \mathsf{curr\text{-}sit} \sqsubseteq \mathsf{flooding}$$

Intuitively we can give a visual account of the argumentation related to the problem (cf. Figure 3). For the linear representation of very we use:

$$\mathsf{very(x)} \quad = \quad \begin{cases} \frac{2}{3}x & : \quad 0 < x < 0.75 \\ 2x - 1 & : \quad 0.75 \leq x \leq 1 \end{cases}$$

As as solution we obtain a support of .33 for the degree to which the description of the current situation is subsumed by the concept *flood.* We argue that this kind of request may be a typical case concerning the knowledge based support of context-awareness.

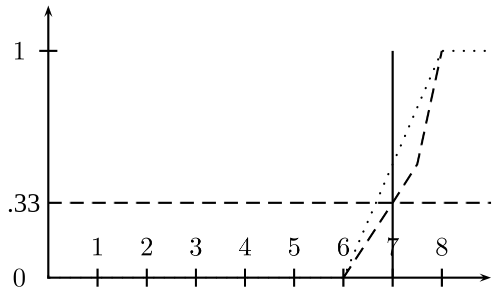

Figure 3: (Very) High.

$$\mathsf{agent} \doteq \exists\, \mathsf{water\text{-}level}. =_{0.7}$$

Figure 4: Terminological Representation of an Agent's State.

3 KNOWLEDGE

In our approach a system's current state is represented in terms of knowledge and its location. On this background we can demonstrate the influences of knowledge on the agents actions and the effects of these actions on the distribution of knowledge. Considering our running example again it becomes clear that knowledge about the environment (e.g. communication paths (or *channels*) has to be available when initiating a communication act. As we already saw in Subsection 2.1 this knowledge may located in the agent or in the environment. In this section we discuss the question how such knowledge is represented and in which way it can be accessed.

Agents. In our approach agents are modeled as P-systems. Thus their state is represented by regions and their knowledge by floating molecules. Since we conceive these terms as assertions related to an ontology we can use predefined complex concepts in order to represent an agent's state. In Figure 4 we show a simple representation of an agents state. In this case the agent's state is characterized by the awareness concerning the water level.

Note that this kind of representation is not accessible for our observation-based approach in which agents are treated as black-boxes. Following this maxim of strict encapsulation we have to rely exclusively on the observation functions *know* and *next* in order to get information about knowledge and behavior. Note that this approach supports the formal treatment of highly abstract characteristics of behavior which are independent of individual details of certain agents.

In order to retrieve some information about the internal state of the agent we have to use the observation function *know* in a way which is very similar to requests to knowledge bases.

$$\langle \mathsf{know}_{agent}(\exists \mathsf{water\text{-}level.High}), 0.33 \rangle$$

$$\begin{aligned}
\mathsf{s}_0 &\doteq \mathsf{Communicate}\\
\mathsf{s}_1 &\doteq \mathsf{Conn\text{-}by\text{-}Conv}\\
\mathsf{s}_2 &\doteq \mathsf{Conn\text{-}by\text{-}Mobile}\\
\mathsf{s}_3 &\doteq \mathsf{Msg\text{-}Sent}\\
\mathsf{s}_4 &\doteq \mathsf{Msg\text{-}Sent}
\end{aligned}$$

$$\begin{aligned}
\mathsf{s}_0 &\sqsubseteq \exists \mathsf{select\text{-}conv.s}_1\\
\mathsf{s}_0 &\sqsubseteq \exists \mathsf{select\text{-}mobile.s}_2\\
\mathsf{s}_1 &\sqsubseteq \exists \mathsf{send.s}_3\\
\mathsf{s}_2 &\sqsubseteq \exists \mathsf{send.s}_4
\end{aligned}$$

Figure 5: Terminological Description of States and Transitions.

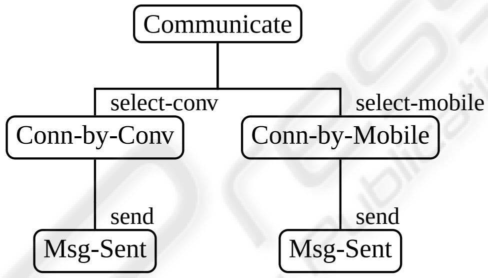

Figure 6: Example: Behavioral Description.

4 BEHAVIOR

For the representation about behavior we exploit the correspondence between description logics and propositional dynamic logic (first published by (Schild, 1991)). This allows us to reuse the same syntactic concepts (which we used for knowledge representation in Section 3) for description of behaviors. Intuitively we use concepts for the description of *states* while roles represent *events* which initiate state transitions. As an example we give a simplified description of the behavior in our example in Figure 5.

In order to support intuitive reasoning about behavioral description we rely on simple a visual notation (cf. Figure 6). In this notation the nodes of a tree represent states. The soft boxes contain (possibly complex) concepts which have to hold in the particular state. Edges are labeled with names of events (represented by role names).

Conformance. When reasoning about systems behavior an essential question is whether a given agent is able to *conform* to a behavioral description. Again

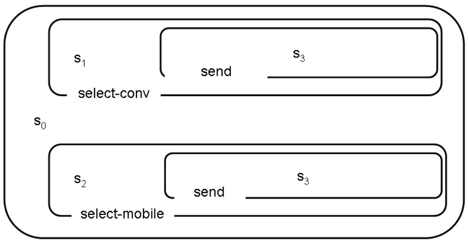

Figure 7: Behavior Represented by Nested Membranes.

we are not interested in retrieving the complete state description or all capabilities of a given agent but exclusively in the question if he is able to perform an action which is described by a terminology. In order to reason about problems of behavioral conformance we introduce a *tree-automaton* which interprets behavioral descriptions as simulation experiments concerning a certain agent (following (Vardi, 1998)). As may be expected the functions *know* and *next* play a prominent role in this experiments.

Since we rely on concepts from membrane computing again we first show how to transform tree-like behavioral descriptions into membrane-based representations (cf. Figure 7). For simplicity we restrict our attention to finite trees with a branching factor equal or less than two. The concepts $s_1, \ldots, s_4$ which were defined in Figure 5 are now represented as molecules floating in solutions contained in membranes. Successor states (of a given state) are represented by embedded membranes which are labeled with a role name representing the event whose occurrence is necessary to access the state.

In order to support conformance testing it has to be checked whether a given agent Ag supports a (possibly incomplete) description of behavior. Intuitively in our observation-based approach it is checked whether the relevant concepts can be observed in the agent's current state and whether the concepts related to the successor states can be observed in the the agent's successor states (which are accessed by the function *next*). The relevant part of this line of reasoning can be described by the following rule.

$$\begin{array}{l}[\mathbf{know}(Ag, A), [_a A[_b B]_b[_c C]_c]_a] \quad \rightarrow \\ \;[\mathbf{know}(\mathbf{next}(Ag, b), B), [_b B]_b], \\ \;[\mathbf{know}(\mathbf{next}(Ag, c), C), [_c C]_c], \\ \quad \mathbf{when\ know}(Ag, A) = \mathbf{true}\end{array}$$

Due to space limitations we only consider a very simple and general case. If the complex concept A which has to be satisfied in the current state of Ag holds in the current state (indicated by $\mathbf{know}(Ag, A) = \mathbf{true}$) all embedded membranes (representing the successor states) are investigated concurrently. For this sake the automaton creates several copies of itself (one for each successor state). For the description of this self-reproductive behavior of the automaton we exploit the feature of membrane division which is described by (Păun, 2002). Note that in this computational model large amounts of computation space can be provided in linear time which makes this type of computation surprisingly efficient. On the right-hand side of the rule each copy of the automaton works on a successor state of A (reached by Ag's transition function *next* together with role names b resp. c) and a subtree of the original behavioral description. For the sake of our example concept names A (resp. B and C) have to be bound to s_0 (resp. s_1 and s_2) while the role names b and c have to be instantiated with *select-conv* and *select-mobile*. In this case a would represent the empty role ϵ.

5 DISTRIBUTED ACTION

As we already stressed we are interested in the support of reasoning about very abstract descriptions of systems behavior. Such incomplete specifications support a style of reasoning which exclusively considers relevant characteristics of behavior (and neglects unimportant details). In this Section we give an example for the treatment of incomplete specifications of distributed actions.

Again we consider our example concerning the choice of communication channels in a highly dynamic scenario (e.g. disaster management). Our discussion in this section is based on the intuition that for our purposes it does not matter which agent decides about the selection of the channel (i.e. the agent himself or the environment). We are interested in supporting a style of reasoning which is general enough to support both scenarios from Subsection 2.1 (i.e. which is general enough to recognize the naive scenario and the presence of an intelligent environment).

In order to support this style of reasoning we simply extend our behavioral description from Section 4 with the following definitions. After this we extend our automaton with the rules which are necessary to process these constructs.

$$\begin{array}{rcl}\textsf{select-conv} & \doteq & \textsf{sel-conv-ag} \sqcup \textsf{sel-conv-env} \\ \textsf{select-mobile} & \doteq & \textsf{sel-mob-ag} \sqcup \textsf{sel-mob-env} \\ \textsf{send-msg} & \doteq & \textsf{snd-msg-ag} \sqcap \textsf{transfer-env}\end{array}$$

Note that in these terminological descriptions we rely on *complex roles* for the description of complex actions. In this case we again exploit the correspondence between the description logics $\mathcal{ALCFI}_{reg}$ and converse-PDL. For example for an occurrence of an event *select-conv* it suffices that the event *sel-conv-ag* or the event *sel-conv-env* occurs. For the occurrence of event *send-msg* on the other hand we require the events *send-msg-ag* and *transfer-env* to occur. These constructs are processed by the following rules. Note that we consider a complex agent $C_{Ag} = \langle Ag, Env \rangle$ in this example which consists from an elementary agent Ag and the environment Env.

$$[\mathbf{know_D}(C_{Ag}, A), [_a A[_b B]_b]_a] \rightarrow [\mathbf{know_D}(\mathbf{next}(C_{Ag}, b_1), B), [_{b_1} B]_{b_1}], [\mathbf{know_D}(\mathbf{next}(C_{Ag}, b_2), B), [_{b_2} B]_{b_2}], \mathbf{when}\ \mathbf{know_D}(C_{Ag}, A), \mathsf{b} \doteq \mathsf{b_1} \sqcup \mathsf{b_2}$$

$$[\mathbf{know_D}(C_{Ag}, A), [_a A[_b B]_b]_a] \rightarrow [\mathbf{know_D}(\mathbf{next}(C_{Ag}, [b_1, b_2]), B), [_{[b_1,b_2]} B]_{[b_1,b_2]}], \mathbf{when}\ \mathbf{know_D}(C_{Ag}, A), \mathsf{b} \doteq \mathsf{b_1} \sqcap \mathsf{b_2}$$

Note that we treat a disjunction of subevents with a rule which is similar to the rule in Section 4. Again both branches of the disjunctions are processed concurrently by individual automata. In the case of conjunctions on the other hand we formulate our experiment concerning the behavior of C_{Ag} using a multiset of events $[b_1, b_2]$ such that no assumption is made about the order of these events. Note that we use a specific observation function for the compositional agent C_{Ag}. We use the function $know_D$ for the representation of *distributed knowledge* (cf. (Fagin et al., 1996)). Formally a proposition ϕ is distributed knowledge among the agents A_i if it is contained in the unification of all their individual knowledge. In the context of our example the following holds:

$$\mathbf{know_D}(C_{Ag}, A) \Leftrightarrow \mathbf{know}(Ag, A) \vee \mathbf{know}(Env, A).$$

6 COORDINATION

An essential problem in pervasive system consists in the relation of individual agents and environment. Related questions are: how can an agent determine if his actions make sense w.r.t. the behavior of the global system? How can the consistency between individual information and the global state of knowledge be ensured? From a systemic viewpoint this problem is often called the *scale gap* (Krasnogor et al., 2005) which may be observed between the behavior of the individual and the strategy of the population. A typical scenario which illustrates the relevance of this issue consists in a sudden and unexpected change of environmental conditions. Obviously it is necessary to distribute information concerning the new state of affairs as fast as possible. Especially it is not adequate to distribute large quantities of information in such situations. In fact it is prefered to distribute the message that *something happened* as fast as possible. This is necessary in order to prevent the agents from following strategies and routines which are inadequate given the new environmental conditions.

We argue that the observation of social team performance as well as the adoption of biological coordination mechanism can contribute to the solution of this problem. As a starting point we take the observation that in critical situations (which are frequently caused by environmental changes) human agents do not rediscuss all relevant topics. In fact they use they use inexpensive communication mechanisms (e.g. nonverbal communication) in order to distribute the relevant changes as fast as possible.

As a matter of fact such coordination mechanisms have been observed in biology. Thus *quorum sensing* is used by bacteria in order to coordinate individual behavior and the strategy of the population. In our proposal we use the computational models which were defined for this coordination mechanism using P-systems. Especially we apply simplified versions of the concepts described in (Krasnogor et al., 2005). We claim that the application of these models is helpful for the understanding of social coordination and for the enhancement of coordination in pervasive settings.

Basic Mechanisms. Since we are not able to describe the overall model (due to space limitations) we have to content ourselves with the discussion of some basic mechanisms. Biological coordination heavily relies on the production and distribution of *signal molecules* which are collected in the environment in order to reflect the actual global state. Agents can receive such molecules and draw inferences about the situation.

Example. We claim that in complex systems information about some facts is distributed all over the system by mechanisms which are subtle but highly efficient. For the sake of modeling we reduce these mechanisms to knowledge diffusion. We claim that agents produce information which is floating in the environment while executing cooperative tasks. For example an agent which detects that the an environmental conditions (e.g. the water level) deteriorates distributes some information about this observation while concurrently performing his tasks. We can model this simple behavior by a rule which generates information molecules.

$$[\mathsf{water\text{-}level\text{-}high}] \rightarrow [(\mathsf{water\text{-}level\text{-}high}, \mathsf{here}), (\mathsf{serious\text{-}event}, \mathsf{out})]$$

While the molecule *water-level-high* represents a fuzzy assertion concerning the water level the molecule *serious-event* is considered as a global stress indicator. Note that this stress indicator is generated repeatedly (as long as the water level is considered as critical). Obviously the concentration of this type of molecule in the environment is significant. Since the environment's state is represented by a multiset the concentration can be represented by the molecules cardinality. The degree to which a certain an agent is susceptible to systemic stress can be modeled by a constant τ which defines a threshold for the stress niveau.

$$[\mathsf{serious\text{-}event}^{\mathsf{p}}] \quad \rightarrow \quad [\mathsf{need\text{-}to\text{-}act}], \mathsf{p} > \tau$$

When a certain concentration of stress is reached agents are urged to act. This alertness is represented my a molecule *need-to-act* which may operate as a catalyzator for further actions. When the situation has stabilized the stress indicator is metabolized by the environment (e.g. using rules like $serious\text{-}event \rightarrow \Lambda$). Note that the molecule *serious-event* represent only very general and vague information w.r.t. the systems global state. This usage of vague information is typical for robust coordination in critical situations.

7 CONCLUSIONS

In this paper we focused on the diffusion of knowledge in complex system and pervasive settings as a major factor shaping global behavior. We argued that a sensitive handling of this floating of information enables new possibilities concerning the understanding and creation of novel kinds of behavior. We introduced concepts from description logics for the representation of knowledge and demonstrated the treatment of highly abstract and incomplete behavioral descriptions. For the formulation of the related algorithms we introduced concepts from membrane computing. Finally we argued that the transfer of sophisticated interactions and coordination mechanisms from fields like biology or sociology is possible on the basis of this paradigm. As direct benefits of such an approach we emphasize increased abilities to provide meaningful behaviors in dynamic environments and pervasive settings. Especially features like context-awareness and autonomic behavior are supported by this knowledge-based approach. Although we think that the concepts from membrane computing are a good foundation for modeling and simulation their full computational power could solely exploited on non-conventional hardware. We plan to extend our research in this direction in the future.

We are indebted to the anonymous reviewers for their lucid comments on an earlier versions of this paper.

REFERENCES

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., editors (2003). *The Description Logic Handbook*. Cambridge University Press, Cambridge, U.K.

Berry, G. and Boudol, G. (1992). The chemical abstract machine. *Journal of Theoretical Computer Science*, 96(1):217–248.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1996). *Reasoning about Knowledge*. The MIT Press, Cambridge, Mass.

Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1):41–50.

Klir, G. J. and Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice Hall, Upper Saddle River, N.J.

Krasnogor, N., Georghe, M., Terrazas, G., Diggle, S., Williams, P., and Camara, M. (2005). An appealing computational mechanism drawn from bacterial quorum sensing. *Bulletin of the European Association for Theoretical Computer Science*.

Leveson, N. (1995). *Safeware. System safety and computers*. Addison Wesley, Reading, Mass.

Păun, G. (2000). Introduction to membrane computing. *Journal of Computer and System Science*, 61(1):108–143.

Păun, G. (2002). *Membrane Computing. An Introduction.* Springer, Berlin.

Pepper, P., Cebulla, M., Didrich, K., and Grieskamp, W. (2002). From program languages to software languages. *The Journal of Systems and Software*, 60.

Perrow, C. (1984). *Normal Accidents. Living with High-Risk Technologies*. Basic Books, New York.

Rutten, J. (1996). Universal coalgebra: a theory of systems. Technical report, CWI, Amsterdam.

Schild, K. (1991). A correspondence theory for terminological logics: Preliminary report. Technical Report KIT-Report 91, Technische Universität Berlin, Franklinstr. 28/29, Berlin, Germany.

Straccia, U. (2001). Reasoning with fuzzy description logic. *Journal of Artificial Intelligence Research*, 14:137–166.

Vardi, M. Y. (1998). Reasoning about the past with two-way automata. In *Proc. of the 25th International Colloquium on Automata, Lanugages,and Programming*. Springer.

Wooldridge, M. J. (1992). *The Logical Modelling of Computational Multi-Agent Systems*. PhD thesis, Department of Computation. University of Manchester.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.