

REACTIVE COMMONSENSE REASONING

Towards Semantic Coordination with High-level Specifications

Michael Cebulla

*Institut für Softwaretechnik und Theoretische Informatik, Technische Universität Berlin
Franklinstraße 28/29, 10587 Berlin, Germany*

Keywords: Context-Awareness, Commonsense Reasoning, Semantic Coordination.

Abstract: In contemporary distributed applications questions concerning coordination have become increasingly urgent. There is a trade-off however to be made between the need for a highly reactive behavior and the need for semantically rich high level abstractions. Especially w.r.t. context-aware applications where various systems have to act together and come to coordinated conclusions the need for powerful semantic abstractions is evident. In our argument we start with a calculus for highly reactive behavior. Then we introduce stepwise two extensions w.r.t. the representation of semantic relationships. The first extension concerns the integration of description logics in order to represent statements about the current situation. The main extension however concerns the integration of *classifications* (also known as *formal contexts*). By integrating these highly abstract notions into our membrane-based calculus we make a proposal for the support of common sense reasoning during runtime. The main purpose of the resulting framework is to provide a generic notion of context which is accessible for a rigorous computational treatment during runtime. We claim that this proposal is a contribution to the robustness of systems behavior and context-awareness.

1 INTRODUCTION

In this paper we make an attempt to bridge the gap between highly reactive behavior during runtime and the need for highly abstract and meaningful concepts for context-awareness. Especially we propose to integrate highly abstract forms of commonsense reasoning (Barwise and Seligman, 1997; Ganter and Wille, 1997) with membrane computing (Păun, 2000) in order to support a way of runtime reasoning whose robustness is comparable to human reasoning. By this proposal we extend previous suggestions concerning concepts for high-level and intuitive specifications (Pepper et al., 2002). Our general intent is directed towards a generic treatment of contextual influences during runtime.

While classifications and formal contexts heavily rely on universal algebra and category theory (Goguen, 2005) we propose an operational foundation using membrane computing. In this paper we focus on the treatment of quotients of classifications w.r.t. specific invariants. Quotients are formal con-

structs which provide powerful support for common sense reasoning. Using these mechanisms systems can make inferences about the current situation during runtime. In order to find out whether an invariant holds in the current situation the system has to derive the quotient of the classification describing the current situation and a specific invariant. This invariant then can be considered as a request. If the quotient is not empty the invariant holds.

Related Work. Important and influential treatments of the notion of context are (McCarthy, 1997; Sowa, 2000). Even more rigorous treatments can be found in (Barwise and Seligman, 1997; Ganter and Wille, 1997; Goguen, 2005). Our discussion is basically inspired by (Barwise and Seligman, 1997) shifting the focus to an operational treatment of runtime inferences.

In the following we briefly describe the formal foundations of our approach in Sections 2- 4. Then we describe the notions of invariants and quotients and discuss their algorithmic treatment in our frame-

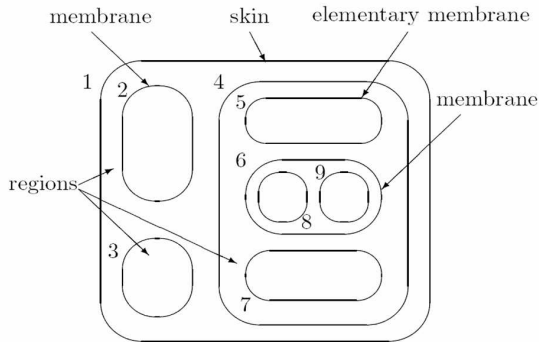


Figure 1: Intuition: Membrane System (Bernardini, 2005).

work. We claim that this approach is quite powerful since it supports the reasoning about different types of relations in different types of logics. Examples are various types of modal and behavioral logics. In order to give an impression of the power of the approach we take an example from the literature which is quite relevant for questions concerning coordination in distributed systems.

2 MEMBRANE COMPUTING

In this Section we describe membrane-based *P-systems* (Păun, 2000) as building blocks for the description of complex systems. Thus a transition *P-system* is basically defined as a constraint store whose behavior is described by CHAM-like transition rules for multiset rewriting (Banâtre and Metayer, 1993). *P-Systems* are defined on membrane structures as follows (slightly adapted from (Păun, 2000; Bernardini, 2005)):

Definition 1 (P-System) A *P-system* of degree m is defined as a tuple

$$\Pi = \langle V, C, \mu, w_1, \dots, w_m, (R_1, \rho_1), \dots, (R_m, \rho_m), i_0 \rangle,$$

where V is an alphabet of symbols (called objects), $C \subseteq V$ is a subset called catalysts, μ is a membrane structure, w_1, \dots, w_m are fuzzy multisets of objects from V , R_1, \dots, R_m are sets of transformation rules associated with the regions, ρ_i are the priority relations between these rules and i_0 is the output membrane. \square

The symbols of V are treated as molecules floating in a specific solution. These (sub-)solutions are also referred to as *regions* (cf. Figure 1). Regions are associated with a membrane which contains them. Additionally transformation rules are associated to regions which define local behavior.

Adaptivity, Self-Optimization. We chose this highly reactive semantic model as the basis of our process description because we feel that it is appropriate for the description of unexpected behavior. Especially, environmental changes or unexpected contextual influences can be modeled by introducing new molecules into the solution. Thus, the actual state of a *P-system* is described by the states of its regions. The terms in these regions can be processed using the transition rules or can be propagated to other regions by diffusion through porous membranes.

Rules. Rules in *P-systems* are of the following form (Păun, 2000):

Definition 2 (Rules) Rules in *P-systems* are of the form $u \rightarrow v$ with $u \in V^+$ and $v \in (V \times Tar)^*$, where $Tar = \{here, in, out\}$. \square

While u in this kind of rules is a multiset over C , v is a pair whose first element is an element of C while the second element is taken from $\{here, in, out\}$. The latter keywords (also called *target commands*) specify the direction into which v has to be moved. Rules are associated with regions. In each step of the behavior of a *P-system* a transition takes place which consists of the application of all applicable rules in all regions. The rules are applied in a non-deterministic and maximal parallel manner.

Remark. In our discussion we silently introduce a fuzzy version of *P-systems*.

3 FUZZY DESCRIPTION LOGICS

For the fuzzification of description logics fuzzy sets (Zadeh, 1965) are introduced into the semantics instead of the crisp sets used in the traditional semantics cf. (Baader and Nutt, 2003). For more detailed discussion of these issues cf. e.g. (Straccia, 2001; Hödöbler et al., 2003).

Definition 3 (Fuzzy Interpretation) A fuzzy interpretation is a pair $I = (\Delta^I, \cdot^I)$, where Δ^I is, as for the crisp case, the domain whereas \cdot^I is an interpretation function mapping

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

\square

If C is a concept then C^I will be interpreted as the *membership degree function* of the fuzzy concept C w.r.t. I . Thus if $d \in \Delta^I$ is an object of the domain Δ^I then $C^I(d)$ gives us the degree of being the object d an element of the fuzzy concept C under the interpretation I (Straccia, 2001). For some selected constructors which were considered for description logics the interpretation function \cdot^I has to satisfy the following equations:

$$\begin{aligned}
 \top^I(d) &= 1 \\
 \perp^I(d) &= 0 \\
 (C \sqcap D)^I(d) &= \min(C^I(d), D^I(d)) \\
 (C \sqcup D)^I(d) &= \max(C^I(d), D^I(d)) \\
 (\neg C)^I(d) &= 1 - C^I(d) \\
 (\forall R.C)^I(d) &= \inf_{d' \in \Delta^I} \{\max(1 - R^I(d, d'), C^I(d'))\} \\
 (\exists R.C)^I(d) &= \sup_{d' \in \Delta^I} \{\min(R^I(d, d'), C^I(d'))\} \\
 (\forall T.C)^I(d) &= \inf_{d' \in \Delta^I} \{\max(1 - T^I(d, o), C^I(o))\} \\
 (\exists T.C)^I(d) &= \sup_{d' \in \Delta^I} \{\min(T^I(d, o), C^I(o))\} \\
 (qR.C)^I(d) &= \{d' \in \Delta^I, |\{d'' \mid R^I(d, d'') > 0\}| \geq q\} \\
 (\text{mod}_q R.C)^I(d) &= \{d' \in \Delta^I, \text{mod}(\{d'' \mid R^I(d, d'') > 0\}) \geq q\}
 \end{aligned}$$

Quotient-based Reasoning. Deviating from common approaches based on description logics we do not focus on model-based reasoning about satisfiability or subsumption. This is the reason why we do not rely on tableaux-based reasoning and thus do not have to face the resulting computational complexity (Baader and Nutt, 2003). In contrast we propose an automata-based approach for reasoning about the conformance of systems w.r.t. certain invariants. Specifications are considered invariants for which the quotients are computed by tuple automata. Note that we have our restrict our attention to the treatment of crisp invariants in this paper due to space limitations. It is well-known that this type of invariant can be considered as a special case of fuzziness.

4 CLASSIFICATIONS: FORMAL CONTEXTS

In this Section we briefly describe the integration of high-level modeling into the reactive calculus.

4.1 Information Systems

The abstract notion of information systems has been introduced by (Barwise and Seligman, 1997) as abstract description of components in distributed systems.

Information Systems. Information systems are defined by classifications.

Definition 4 (Classification) A classification $\mathbf{A} = \langle \text{tok}(\mathbf{A}), \text{typ}(\mathbf{A}), \models_{\mathbf{A}} \rangle$ is a triple where $\text{tok}(\mathbf{A})$ is a set of tokens (object), $\text{type}(\mathbf{A})$ is a set of types classifying tokens, and $\models_{\mathbf{A}}$ is a binary relation between $\text{tok}(\mathbf{A})$ and $\text{typ}(\mathbf{A})$. \square

The notion of classification can be represented by P-systems. We establish this connection in order to provide an operational semantics for knowledge-based reasoning. We exploit in this representation the similarities between classifications, knowledge bases (e.g. for description logics) and formal contexts. Generally the following components are used (we ignore the terminological differences between the notions from different approaches):

Types. The sets of concepts and of roles constitute the vocabulary of the type language. The alphabet V from the definition of P-systems consists from the set C of concepts and the set R of roles. Intuitively these sets contain the vocabulary (i.e. the signature) which can be used in the current context.

Individuals. Names for individuals which are also part of the TBox in case of knowledge bases are treated as tokens in classifications. In our approach based on P-systems a set of individual names is part of the type language.

Classification Relation. In the case of knowledge bases this relation is described by the expressions in the ABox. Basically it defines which types are applicable to which individuals.

Simulation. Under operational criteria we represent classifications as P-systems P_{CI} . Basically a classification is enclosed by a membrane. On the background of our discussion we distinguish three subsystems in P_{CI} : a signature, a set of axioms, and a set of sentences. These entities are mapped to components from P-systems.

Definition 5 (Classification P-System P_{CL}) The P-System for the representation of classifications P_{CL} is defined by the tuple $\langle V, L, \mu, w_i, R \rangle$ where $V = C \cup R \cup I$ is the terminology of the classification (i.e. the signature or the type set or the TBox), the label set $L = C \cup R$. The initial membrane structure $\mu = [0[A]_A]_0$ contains an empty membrane representing an ABox. \square

Example 1 (Simple Classification.) We use a table for the classification of patients according to gender resulting in the DL-expressions in the bottom part of Figure 2. \diamond

	p_1	p_2	p_3	p_4	p_5
msc	x	x			x
fem			x	x	

$msc(p_1)msc(p_2)fem(p_3)$
 $fem(p_4)msc(p_5)$

Figure 2: Classification of Patients w.r.t. Gender.

The same situation is represented by the following P-system:

Definition 6 (P-system P_{Ex}) The P-System P_{Ex} is defined by the tuple $\langle V, L, \mu, w_i, R \rangle$ where

$$V = \{msc, fem\} \cup \{p_1, p_2, p_3, p_4, p_5\}$$

is the terminology of the classification (i.e. the signature or the type set or the TBox), the label set $L = C \cup R$. The initial membrane structure $\mu = [0[A^V]A]_0$ contains the multiset

$$v = [msc(p_1), msc(p_2), fem(p_3), fem(p_4), msc(p_5)].$$

□

4.2 Derivation, Closure

As is well known from the literature classifications and formal contexts are essentially the same. The concepts (in DL-terminology) which are contained in the type language can be treated as the attributes of formal contexts while individuals take the role of tokens. In FCA-terminology the notions of formal concepts is related to closed theories (Goguen, 2005). In this Subsection a computational approach to some basic inference mechanisms is discussed which is used as a foundation of the quotient-based mechanisms in Subsection 4.3.

Derivation. As a first step we have to describe the computation of the derivation operator (Ganter and Wille, 1997). This operator describes the relation between signatures and models and induces a Galois connection (Goguen, 2005).

Definition 7 (Derivation) For a set $A \subseteq G$ of individuals the derivation is defined by:

$$A' := \{m \in M \mid gIm \text{ for all } g \in A\}.$$

for the set of $B \subseteq M$ of attributes:

$$B' := \{g \in G \mid gIm \text{ for all } m \in B\}$$

where I is the classification relation. □

Again we define this operator by membrane manipulations.

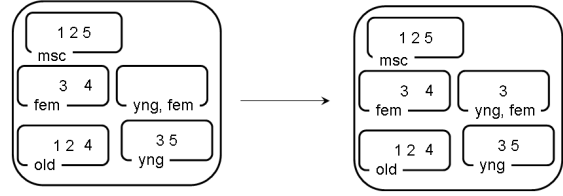


Figure 3: Derivation.

Simulation. Thus for a set of attribute names B all instances have to be collected which satisfy these attribute names in a membrane which is labeled with B' . The resulting membrane is the intersection of all membranes which represent concepts contained in B .

$$\begin{aligned} [B']_{B'}[Ca]_C &\rightarrow [B'a]_{B'}[Ca]_C, C \in B \\ [B'a]_{B'}[Cb]_C &\rightarrow [B'a \cap b]_{B'}[Cb]_C, C \in B \end{aligned}$$

An exemplary computation is shown in Figure 3. The complementary definition for the derivation operator on sets of instances we use a similar procedure which operates on the flipped classification, cf. (Barwise and Seligman, 1997).

4.3 Quotients of Classifications

In this Section we describe the computation of quotients which is very important and opens many possibilities of application. As a running example we describe the reasoning about modalities which can be mapped to the computation of quotients for classifications and invariants. The relation of the invariant then represents the accessibility relation of the Kripke semantics (Hughes and Cresswell, 1996).

Definition 8 (Invariant) An invariant $I = \langle \Sigma, R \rangle$ is a pair of a set of types Σ and a relation R between a set of tokens from the classification. □

Intuitively we claim that the accessibility relation of a modal logics is represented by R while the tokens may be considered as *possible worlds*. The knowledge which is contained in a classification is then expressed by the set of types Σ which holds in every possible world. More formally the knowledge which is contained in a classification is represented by the quotient of the classification.

Definition 9 (Quotient of a Classification) Let $I = \langle \Sigma, R \rangle$ be an invariant on the classification A . The quotient of A by I , written A/I , is the classification with types Σ , whose tokens are the R -equivalence classes of tokens in A , and with $[a]_R \models_{A/I} \alpha (\alpha \in \Sigma)$ iff $a \models_A \alpha$. □

There is a close correspondence between the notion of quotient in (Barwise and Seligman, 1997) and the notion of derivation (of intents) in (Ganter and Wille, 1997). We thus see an intent as a special case of invariant (Barwise and Seligman, 1997). One way to state this correspondence is to claim that an intent is an invariant (with an unspecified relation).

Remark. It is possible to model different kinds of knowledge defining different properties of the relation R (e.g. reflexivity, symmetry and transitivity). As is well known each of these different types of relations induce different types of modal logics (Hughes and Cresswell, 1996). Although the original definition of invariants assumes equivalence relations we decided to admit also other types of relations. In these cases the quotients cannot be considered as equivalence classes anymore.

Example 2 (Quotients of Modalities.) Again we rely on tables in order to specify situations. We use a table for the description of classification and another table in order to describe relations (cf. Figure 4). Again these tables can be easily translated into expressions from description logics. Note that we give only one half of the specification for the relations since we assume that they are symmetric. The situation described in the tables corresponds to the muddy children puzzle as treated by (Fagin et al., 1996). A tabular notation is provided for the easy description of such a situation. The first table describes the set of possible worlds s_i which are characterized by expressions built from the attributes m_i , where m_i denotes the fact that child i has a muddy forehead. The relations r_i described by table 2 on the other hand describes which world is accessible for agent i in a given world.

Invariants. We are able to specify invariants $\langle \Sigma, R \rangle$ on this structure where $\Sigma \subseteq \{m1, m2, m3\}$ and $R \subseteq \{r1, r2, r3\}$. Using the syntax of description logics (and exploiting semantic correspondences to the logic \mathbf{K}_m we can express some interesting properties as invariants .

$$\begin{aligned} \exists r1.m1 &\Leftrightarrow m1 \text{ is possible for } A_1 \\ \forall r1.m1 &\Leftrightarrow m1 \text{ is necessary for } A_1 \end{aligned}$$

By using expressive description logics we are able to describe more complex modal properties.

$$\begin{aligned} \exists(r1 \cup r2).m1 &\Leftrightarrow m1 \text{ is possible for } A_1 \text{ and } A_2 \\ \forall(r1 \cup r2).m1 &\Leftrightarrow m1 \text{ is distributed knowledge.} \\ \forall(r1 \circ r2).m1 &\Leftrightarrow A_1 \text{ knows that } A_2 \text{ knows } m1 \end{aligned}$$

◇

	m1	m2	m3
s1			
s2			x
s3		x	
s4		x	x
s5	x		
s6	x		x
s7	x	x	
s8	x	x	x

	s1	s2	s3	s4	s5	s6	s7	s8
s1								
s2	r3							
s3	r2							
s4		r2	r3					
s5	r1							
s6		r1				r3		
s7			r1			r2		
s8				r1		r2	r3	

Figure 4: Three Modalities.

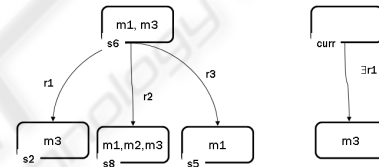


Figure 5: Membrane representation of Kripke structures.

Discussion. By the integration of these three methods in our framework we get an ensemble of intuitive specification methods which are domain-specific and easy to use. Especially for the usage of domain-experts we propose the usage of tables (as is widespread in FCA and IFF-theory).

Simulation. In order to treat relations of invariants in our framework we transform the corresponding expressions into a membrane-based representation. For this sake we consider Kripke structures as flipped classifications (Barwise and Seligman, 1997). These are characterized by the fact that the membranes are labeled by the names of the tokens (i.e. the possible worlds) while the type names are floating in the solution. Intuitively each membrane contains the types which are satisfied in the world whose name is used as a label for the membrane. Relations between the worlds are represented by membrane channels between the membranes.

For the algorithmic treatment we extensively use

the fact that we can consider specifications (classifications as well as invariants) as tree-like term structures. For example both the systems specification as well as the viewpoint specifications can be represented in such a way (cf. Figure 5). Thus the quotient can be computed by a simple tuple tree automaton which can be defined using rules of P-systems. For the sake of brevity we have to skip the details of the algorithmic treatment.

5 CONCLUSION

Our research is directed towards an integration of highly reactive behavior on one hand and the support of common sense reasoning which relies on powerful semantic abstractions on the other hand. In this paper we proposed an approach which contains notions from information flow, formal concept analysis, description logics and membrane computing in order to attain this goal. In this paper the specific contribution is represented by the introduction of classifications, invariants and quotients into the calculus of membrane computing. This can be considered as a foundation for the integration of more complex abstractions.

REFERENCES

- Baader, F. and Nutt, W. (2003). Basic description logics. In Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., editors, *The Description Logic Handbook*. Cambridge University Press, Cambridge, U.K.
- Banâtre, J. and Metayer, D. L. (1993). Programming by multiset transformation. *Communications of the ACM*, 36:98–111.
- Barwise, J. and Seligman, J. (1997). *Information Flow: the logic of distributed Systems*. Cambridge Univ. Press.
- Bernardini, F. (2005). *Membrane Systems for Molecular Computing and Biological Modelling*. PhD thesis, University of Sheffield.
- Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1996). *Reasoning about Knowledge*. The MIT Press, Cambridge, Mass.
- Ganter, B. and Wille, R. (1997). *Formal Concept Analysis. Mathematical Foundations*. Springer: Berlin, Heidelberg.
- Goguen, J. (2005). Information integration in institutions. In *Memorial Volume for Jon Barwise*. Indiana University Press, Indiana.
- Höbdober, S., Störr, H.-P., and Khang, T. D. (2003). The fuzzy description logic alc_{FH} with hedge algebras as concept modifiers. *International Journal of Advanced Computational Intelligence and Intelligent Informatics*, 7(3):294–305.
- Hughes, G. and Cresswell, M. (1996). *A New Introduction to Modal Logic*. Routledge.
- McCarthy, J. (1997). Formalizing context (expanded notes). In Aliseda, A., van Glabbeek, R., and Westerstahl, D., editors, *Computing Natural Language*. Stanford University.
- Păun, G. (2000). Introduction to membrane computing. *Journal of Computer and System Science*, 61(1):108–143.
- Pepper, P., Cebulla, M., Didrich, K., and Grieskamp, W. (2002). From program languages to software languages. *The Journal of Systems and Software*, 60.
- Sowa, J. F. (2000). *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks/Cole, Pacific Grove.
- Straccia, U. (2001). Reasoning with fuzzy description logic. *Journal of Artificial Intelligence Research*, 14:137–166.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.